UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

Chair of Practical Computer Science IV:
Dependable Systems Engineering

Master's Thesis

# Vulnerabilities in Privacy-Preserving Record Linkage: The Threat of Dataset Extension Attacks

as part of the degree program Master of Science Business Informatics submitted by

Marcel Mildenberger
Matriculation number 1979905

on August 1, 2025.

Supervisor: Prof. Dr. Frederik Armknecht
PhD Student Jochen Schäfer

# Abstract

Privacy-Preserving Record Linkage enables dataset integration across institutional boundaries without disclosing Personally Identifiable Information through similarity-preserving encoding schemes such as Bloom Filter, Tabulation MinHash, and Two-Step Hash. However, these schemes are vulnerable to inference-based attacks that exploit structural properties of encoded data. This thesis introduces the Dataset Extension Attack, a novel two-stage approach that extends the capabilities of the Graph Matching Attack by enabling the re-identification of individuals who were previously unmatched.

The Dataset Extension Attack leverages Artificial Neural Network to learn statistical relationships between encoded representations and underlying plaintext n-grams. Training on re-identified records from a preceding Graph Matching Attack, the Dataset Extension Attack generalizes to infer n-gram distributions for previously unmatched entries. The attack is framed as multi-label classification, where the Artificial Neural Network predicts n-gram presence in original identifiers based on encoded representations. Three reconstruction strategies are explored: graph-based reconstruction, dictionary-based fuzzy matching, and Large Language Model-based semantic reconstruction.

Extensive experiments using synthetic and semi-realistic datasets evaluate the Dataset Extension Attack across varying dataset sizes, encoding schemes, and overlaps. Results show that the Dataset Extension Attack significantly outperforms frequency-based baselines, achieving over 28% re-identification in favorable configurations and highlighting substantial privacy risks.

Key findings reveal that Bloom Filter and Two-Step Hash expose recurring patterns exploitable through Artificial Neural Networks, while Tabulation MinHash shows highest resilience though still permitting partial reconstruction. A non-linear correlation exists between prediction quality and re-identification success, with attacks becoming increasingly effective as Artificial Neural Network F1-scores exceed 0.9.

This thesis contributes a scalable attack model highlighting structural vulnerabilities in widely deployed Privacy-Preserving Record Linkage schemes, underscoring the need to reassess current protocols and develop encoding mechanisms that resist statistical inference.

# Contents

# List of Figures

# Acronyms

# 1 Introduction

Linking data and records is an important component of research, software development and software projects. The primary reason for integrating data from different sources is to gain richer, more comprehensive insights about the same entity. Initially, deterministic record linkage, which relies on exact matches between predefined identifiers such as unique IDs, was the main method used in early linkage techniques. However, deterministic approaches often fail in real-world scenarios where data may suffer from inconsistent formatting, typographical errors or missing values, making exact matches impossible [HSW07].

The introduction of a probabilistic framework for record linkage by Fellegi and Sunter in 1969 [FS69] marked an advancement in overcoming the limitations of earlier deterministic approaches. In their seminal work, "A Theory for Record Linkage", they proposed a statistical model that calculates the probability that two records refer to the same entity, even in the presence of inconsistencies or missing data. The model evaluates agreement and disagreement patterns across selected attributes and assigns weights based on the likelihood of a true match or non-match. By systematically accounting for real world data variability, the Fellegi–Sunter model has become a foundational methodology for data linkage, particularly in heterogeneous or distributed environments where exact matching is often infeasible [FS69].

Such a probabilistic approach to record linkage is important in sectors such as healthcare and social sciences, where data is often distributed across multiple institutions or sources and lacks unique identifiers. In these fields, the ability to integrate datasets is essential for gaining insights and improving outcomes. In the United States, for example, the healthcare system is highly fragmented, consisting of numerous independent entities such as hospitals, clinics, insurance companies, public health agencies, and research institutions. Each of these organisations collects and stores patient data independently, often using different systems and, more importantly, different formats. This fragmentation creates challenges when trying to track patient outcomes, monitor disease outbreaks, or evaluate the effectiveness of treatments across populations. Effective data linkage can bridge these gaps by linking records that refer to the same individual across multiple datasets. This integration is important for tasks such as epidemiological research, public health surveillance, personalised medicine and healthcare quality improvements [PSZ+24; VSCR17].

For example, during the COVID-19 pandemic, the inability to efficiently link data between testing centres, hospitals and vaccination sites limited timely tracking of infection rates and vaccination outcomes. Had more robust data linkage mechanisms been in place, public health officials could have responded more effectively to outbreaks and targeted interventions to specific populations. Data linkage thus plays an important role in transforming fragmented data landscapes into unified and actionable insights, leading to more informed decision-making. In response to the COVID-19 pandemic, organisations such as the Centers for Disease Control and Prevention and the Food and Drug Administration have launched projects to address these challenges and further develop linkage techniques [PSZ+24].

In scenarios such as the COVID-19 pandemic, data integration efforts often involve linking records of natural persons from multiple sources. For example, integrating data from differ-

ent healthcare providers, laboratories and public health agencies typically requires the use of pseudo-identifiers derived from Personally Identifiable Information, such as names, dates of birth or other sensitive information. However, reliance on Personally Identifiable Information for linkage raises privacy concerns, as improper handling of such data can lead to re-identification of individuals, with potentially serious consequences such as data breaches, identity theft or unauthorised access to personal health information [PSZ+24; SBR09].

The increasing digitisation of personal data has already led to large-scale data breaches, demonstrating the risks of improperly secured data. Notable incidents such as the Cambridge Analytica scandal, in which personal data was misused for political profiling, highlights the ethical and regulatory challenges of data integration [IH18]. Similarly, healthcare data leaks have raised concerns about the implications of unauthorised access to medical histories, genetic data and insurance records. Leaks of personal health information can have serious consequences, including blackmail, discrimination, and fraud, which can cause personal harm. For example, individuals whose medical histories are exposed may face discrimination in employment or insurance, while others may become targets of scams that exploit their health conditions. The potential for such abuse underlines the critical importance of robust data protection measures by working with Personally Identifiable Information [Smi16].

To address these privacy risks, various techniques have been developed to protect Personally Identifiable Information during the linking process, primarily by encoding the data prior to linking. However, the use of encoded Personally Identifiable Information as pseudo-identifiers presents additional challenges. The key question is how to efficiently encode sensitive information while maintaining the ability to accurately match records [SBR09].

Therefore, Privacy-Preserving Record Linkage techniques are designed to facilitate data integration without exposing sensitive information, ensuring that datasets can be securely linked across different entities. To enable linkage while preserving privacy, similarity preserving encoding is applied to the Personally Identifiable Information. Without such similarity preserving encoding, matches between encoded entities in different databases would not be possible [SBR09; VSCR17]. Over time, three main privacy-preserving encoding schemes have emerged as enablers for Privacy-Preserving Record Linkage [SAH24; VCRS20].

Bloom Filter encoding is the most widely used technique in Privacy-Preserving Record Linkage and is often considered the reference standard [SAH24]. Originally introduced by Burton Bloom in 1970 as a probabilistic data structure for efficient set membership testing [Blo70], Bloom Filters were later adapted for Privacy-Preserving Record Linkage due to their simplicity and efficiency in both storing and computing set similarities. Their compact representation and probabilistic nature make them ideal for scalable Privacy-Preserving Record Linkage systems, especially in environments dealing with large datasets [SBR09]. The seminal work of Schnell et al. [SBR09] demonstrated the use of Bloom Filters in Privacy-Preserving Record Linkage, particularly in healthcare, highlighting their ability to perform secure record matching without exposing sensitive identifiers [SBR09].

However, Bloom Filters are not without limitations. Their vulnerability to graph based attacks and pattern exploitation has driven research into improving their security. Techniques such as diffusion have been proposed to obscure recognisable patterns and increase security [AHS23; SAH24]. For example, Armknecht et al. [AHS23] explored methods to strengthen the security of Bloom Filter by adding a linear diffusion layer to the Bloom Filter based Privacy-Preserving Record Linkage approach, which complicates pattern mining attacks.

To address some of these weaknesses of Bloom Filters, Tabulation MinHash encoding has been introduced as a more secure alternative. MinHash, first developed by Broder in 1997 for

estimating set similarities in large document collections [Bro97], has been adapted using tabulation based hashing [Smi17]. Although less widely used than Bloom Filters, Tabulation MinHash offers distinct advantages, including stronger security guarantees against re-identification attacks. However, these benefits come at the cost of increased computational complexity and memory usage, which may limit its applicability in resource-constrained environments [Smi17].

A further development in encoding techniques is the introduction of Two-Step Hash encoding, which aims to combine the strengths of both Bloom Filters and Tabulation MinHash while mitigating their respective weaknesses. As detailed by [RCS20], Two-Step Hash employs a two-stage process. Data is first encoded using multiple Bloom Filters, followed by an additional hashing layer that transforms the encoded data into a set of integers suitable for similarity comparison. This layered approach enhances privacy by adding an extra layer of obfuscation, making it more resistant to attacks, while maintaining efficient similarity computations [RCS20; VCRS20].

In practice, Bloom Filter based Privacy-Preserving Record Linkage has become the dominant standard and is widely used in areas such as crime detection, fraud prevention and national security due to its balance of efficiency and ease of implementation. However, Bloom Filter based Privacy-Preserving Record Linkage systems are not without limitations and vulnerabilities. Previous research has shown that there are several attacks targeting Privacy-Preserving Record Linkage systems, with a focus on exploiting the weaknesses inherent in Bloom Filter encodings. These attacks specifically target weaknesses in Bloom Filter constructions, such as the weaknesses introduced by possible double hashing, structural flaws in filter design, and susceptibility to common pattern-mining techniques. Notably, no specific attacks have been developed for Tabulation MinHash or Two-Step Hash encodings, suggesting that research has focused primarily on the more widely used Bloom Filter scheme [VCRS20].

However, a more recent and practical attack has emerged that exploits vulnerabilities common to all Privacy-Preserving Record Linkage encoding schemes. The Graph Matching Attack uses publicly available data, such as telephone directories, to re-identify encoded individuals based on overlapping records between plaintext and encoded databases [SAH24; VCRS20]. Unlike previous attacks that focus solely on the encoding scheme of Bloom Filters, the Graph Matching Attack works independently of the encoding scheme chosen. It therefore exploits the graph structure of encoded datasets to re-identify records. Given two datasets, a plaintext reference dataset and an encoded dataset, an attacker can construct similarity graphs where nodes represent individuals and edges represent similarity scores. By solving a graph isomorphism problem, attackers can infer one-to-one mappings between encoded and plaintext records, effectively breaking the privacy guarantees of Privacy-Preserving Record Linkage. The effectiveness of Graph Matching Attacks depends on the overlap between the two sets of data. The greater the overlap, the higher the probability of successful re-identification. While Graph Matching Attacks can successfully re-identify individuals present in both the plaintext and encoded datasets, their effectiveness is limited to the overlapping subset of the two databases [SAH24; VCRS20].

This work aims to go beyond traditional Graph Matching Attacks by re-identifying not only individuals present in the overlapping datasets, but as many individuals as possible from the encoded Privacy-Preserving Record Linkage data. To achieve this, the newly introduced Dataset Extension Attack builds on the foundations laid by Graph Matching Attacks. The Dataset Extension Attack uses an Artificial Neural Network trained on the subset of previously re-identified individuals to predict and decode the remaining encoded records. In doing so, the Dataset Extension Attack expands the scope and effectiveness of the attack, enabling broader

de-anonymisation of Privacy-Preserving Record Linkage datasets beyond the limitations of existing graph based methods.

## 1.1 Motivation

The increasing use of Privacy-Preserving Record Linkage in highly sensitive areas such as healthcare, finance and national security requires research to validate existing techniques and ensure robust privacy [SBR09]. As data-driven applications continue to evolve, the complexity and volume of data being collected and linked across multiple sources is growing rapidly. While Privacy-Preserving Record Linkage systems are designed to facilitate secure data integration without compromising privacy, evolving cybersecurity threats and attack techniques highlight the urgent need to reassess the resilience of these systems [VSCR17].

Privacy has always been a critical concern in data management, but its importance has been intensified in the era of Artifical Intelligence and Machine Learning. These technologies increasingly rely on large data sets, often containing sensitive Personally Identifiable Information such as medical records, financial transactions or behavioural data for training. If compromised, the exposure of such data can lead to privacy violations, including identity theft, financial fraud and discrimination. The rise of data brokerage, where personal information is collected, aggregated and sold, often without explicit user consent, further exacerbates privacy concerns. This commoditisation of personal data has made Personally Identifiable Information an attractive target for malicious actors, increasing the risk of unauthorised data linking and re-identification attacks. As Machine Learning models become more advanced, the demand for rich, high-quality data continues to grow, making privacy an increasingly pressing issue [KM24; MK19].

In this context, the vulnerability of Privacy-Preserving Record Linkage systems to emerging attack methods is of particular concern. While Privacy-Preserving Record Linkage techniques such as Bloom Filter are designed to hide sensitive identifiers during the data linkage process, recent research has shown that these systems are vulnerable to Graph Matching Attacks. Graph Matching Attacks exploit the similarity preserving properties of common encoding schemes to re-identify individuals by comparing patterns in encoded records with those in publicly available plaintext records. This approach undermines the fundamental goal of Privacy-Preserving Record Linkage, to protect sensitive data during the record linkage process. Although Graph Matching Attacks are limited to re-identifying individuals present in both the encoded and plaintext datasets, even partial data exposure in highly sensitive areas can have consequences [SAH24; VCRS20].

The introduction of Dataset Extension Attacks poses an even greater threat to the integrity of Privacy-Preserving Record Linkage systems. Unlike Graph Matching Attacks, Dataset Extension Attacks aim to extend the scope of re-identification to as many individuals as possible within the encoded database. Using Artificial Neural Networks trained on previously decoded data from Graph Matching Attacks, Dataset Extension Attacks can predict and decode additional records, potentially leading to the further deanonymisation of encoded records. This represents a paradigm shift, as it challenges the viability of widely used Privacy-Preserving Record Linkage techniques, such as Bloom Filter based encoding, which have been considered secure.

The primary motivation for this research is to proactively explore and empirically demonstrate the risks posed by advanced inference attacks, with the goal of mitigating their realiza-

tion in real world applications. By revealing the potential vulnerabilities inherent in current Privacy-Preserving Record Linkage systems, this work illustrates how adversaries could exploit partially decrypted or encoded data to compromise individual privacy at scale. A successful implementation of the Dataset Extension Attack provides concrete evidence that existing state-of-the-art methods lack sufficient resilience, thereby underscoring the urgent need for more robust and secure privacy-preserving techniques.

Furthermore, there is a notable gap in current research regarding the extension of attack capabilities beyond the intersection of datasets. While efforts have been made to address the vulnerabilities exposed by Graph Matching Attacks, there is a lack of comprehensive studies exploring how Machine Learning can be used to generalise these attacks and compromise further records. This research aims to fill this gap by developing and evaluating the Dataset Extension Attack, thereby contributing to a broader understanding of Privacy-Preserving Record Linkage vulnerabilities.

By addressing this gap, this thesis aims to contribute to the body of knowledge on Privacy-Preserving Record Linkage vulnerabilities and serve as a foundation for future research aimed at strengthening these systems. The knowledge gained from this study will help to enable the development of more secure Privacy-Preserving Record Linkage techniques.

## 1.2 Related Work

The study by Vidanage et al. [VCRS20] represents an advancement in the field of Privacy-Preserving Record Linkage through the introduction of a new attack method known as Graph Matching Attack. Their work begins with a comprehensive overview of Privacy-Preserving Record Linkage systems and the similarity preserving encoding techniques commonly used, such as Bloom Filters. The Graph Matching Attack exploits weaknesses in these encoding schemes by exploiting their ability to preserve partial similarity information even after encoding. By constructing similarity graphs from both encoded and plaintext datasets, the Graph Matching Attack solves a graph isomorphism problem to align nodes and successfully re-identify individuals in the encoded dataset using publicly available sources such as telephone directories. This method demonstrates the universal applicability of Graph Matching Attacks across different Privacy-Preserving Record Linkage schemes, and highlights a weakness in systems previously thought to be more robust [VCRS20].

Building on this foundation, Schäfer et al. [SAH24] revisited and extended the work of Vidanage et al. Their contribution lies in a reproduction and replication of the original Graph Matching Attack, during which they identified a flaw, an undocumented pre processing step in the provided codebase that increased the effectiveness of the attack. While this step was originally intended to improve computational performance, it introduced errors into the proposed Graph Matching Attack. Schäfer et al. [SAH24] corrected this problem and further optimised the Graph Matching Attack, resulting in improved robustness and efficiency. Their improved implementation achieved higher re-identification rates compared to the original approach. These improvements not only validate the vulnerabilities highlighted by the Graph Matching Attack, but also highlights the potential for refining attack methodologies to expose even greater weaknesses in Privacy-Preserving Record Linkage systems.

The work of Schäfer et al. [SAH24] is particularly relevant to this thesis, as their improved Graph Matching Attack implementation and accompanying codebase form the basis of the Dataset Extension Attack proposed in this study. While the Graph Matching Attack is lim-

ited to re-identifying individuals present in both encoded and plaintext datasets, the Dataset Extension Attack seeks to extend the scope of re-identification beyond this intersection. Using Artificial Neural Networks trained on the re-identified individuals from the Graph Matching Attack, the Dataset Extension Attack aims to predict and decode additional records, potentially leading to complete de-anonymisation of encoded datasets.

To date, no existing research has proposed an approach comparable to the Dataset Extension Attack. This thesis addresses this gap by developing and evaluating the Dataset Extension Attack, thereby contributing to a broader understanding of Privacy-Preserving Record Linkage vulnerabilities and highlighting the urgent need for more secure data linking techniques.

## 1.3 Contribution

The contribution of this thesis is divided into three main parts. First, a comprehensive analysis of Privacy-Preserving Record Linkage systems is carried out, with particular emphasis on the three main encoding schemes: Bloom Filter encoding, Two-Step Hash encoding and Tabulation MinHash encoding. This analysis aims to highlight the basic principles, strengths and weaknesses of each encoding scheme, setting the stage for the subsequent investigation of their susceptibility to the Dataset Extension Attack.

Next, the current state of the art Graph Matching Attack is analysed and its limitations are discussed in detail. Although Graph Matching Attacks have proven effective in re-identifying individuals within overlapping datasets, their applicability is limited to the intersection of plaintext and encoded records. This limitation highlights the need for more advanced attack strategies that can go beyond this.

The main focus of this thesis is the implementation and evaluation of the Dataset Extension Attack, which attempts to outperform the capabilities of Graph Matching Attacks by decoding a larger fraction of encoded records. To achieve this, this thesis examines the conceptual foundations, theoretical underpinnings and technical requirements of the Dataset Extension Attack. Building on the initial re-identifications made by the Graph Matching Attack, the Dataset Extension Attack employs a supervised machine learning approach, specifically using Artificial Neural Networks trained on previously decoded data to predict and re-identify remaining encoded records. This method extends the scope of de-anonymisation in Privacy-Preserving Record Linkage systems and provides a novel approach to current research.

The Dataset Extension Attack is then evaluated against the three main Privacy-Preserving Record Linkage encoding schemes. While the specific encoding scheme has minimal impact on the Graph Matching Attack, which is primarily based on solving a graph isomorphism problem, it plays a role in the Dataset Extension Attack. This is due to the fact that the Artificial Neural Network has to be trained separately for each encoding scheme to account for the unique structural features and nuances of the encoding. However, the Dataset Extension Attack is designed with adaptability in mind, ensuring that it can be applied across different encoding schemes, thus increasing its generalisability and practical relevance.

Through this research, this thesis aims to answer questions about the robustness of Privacy-Preserving Record Linkage systems. It investigates how effective supervised Machine Learning based Dataset Extension Attacks are at re-identifying the remaining entries that Graph Matching Attacks cannot decode. It also examines how different encoding schemes affect the performance and accuracy of the Dataset Extension Attack, providing insight into which schemes are more susceptible to such attacks and why. By addressing these issues, this thesis contributes

to a deeper understanding of the vulnerabilities inherent in Privacy-Preserving Record Linkage systems and lays the groundwork for the development of more secure privacy preserving techniques.

## 1.4 Organization of this Thesis

This thesis is divided into four main sections: technical background, methodology, results, and conclusion.

First, an overview of Privacy-Preserving Record Linkage systems is given, with particular emphasis on a thorough analysis of the most commonly used encoding techniques. Next, the existing Graph Matching Attack is introduced and explained in order to provide the basis for the study. In addition, an overview of Artificial Neural Networks is given to provide the necessary background knowledge.

Next, a detailed description of the attack model for the Dataset Extension Attack is outlined, including how Artificial Neural Networks are used to enhance the attack. This is followed by an explanation of the actual implementation of the Dataset Extension Attack, along with a discussion of the experiments conducted. The results of the Dataset Extension Attack on different encoding schemes are then analysed and evaluated. Finally, this thesis concludes with a summary of the main contributions, a discussion of the broader implications, and suggestions for future research.

# 2 Background

This chapter provides an overview of the key concepts relevant to this thesis, including Privacy-Preserving Record Linkage, various encoding techniques used in secure linkage and existing attacks on encodings and Privacy-Preserving Record Linkage schemes. Privacy-Preserving Record Linkage enables different organizations to link records belonging to the same individual across datasets while preserving privacy. However, the security of such methods can depend on the encoding techniques used to transform sensitive data into a more privacy-preserving format [VCRS20].

To understand the vulnerabilities of Privacy-Preserving Record Linkage systems, we examine three key encoding techniques: Bloom Filter, Tabulation MinHash, and Two-Step Hash, each offering different trade-offs between efficiency, privacy, and robustness against attacks. While these methods aim to prevent direct access to plaintext identifiers, they remain susceptible to adversarial techniques designed to infer or reconstruct the original data [SAH24; VCRS20]. One such adversarial approach is the Graph Matching Attack, which leverages structural similarities between encoded and non-encoded datasets to re-identify individuals [SAH24; VCRS20].

By integrating these concepts, this chapter establishes the necessary foundation for understanding the Dataset Extension Attack introduced later in this thesis, demonstrating how Machine Learning techniques can be employed to bypass existing privacy-preserving mechanisms.

## 2.1 Overview of Privacy-Preserving Record Linkage

Privacy-Preserving Record Linkage enables the linkage of records from different databases that refer to the same individual while trying to preserve privacy. Traditional record linkage relies on unique identifiers, but these are often unavailable or inconsistent due to variations in formatting, spelling or missing data in a distributed environment. Linking records directly on plaintext data poses high privacy risks. To mitigate these risks and prevent further threats to individuals, regulations such as the European Union's General Data Protection Regulation have been established to govern the handling of Personally Identifiable Information [SAH24; VCRS20].

To enable record linkage using Personally Identifiable Information while trying to preserve privacy, Privacy-Preserving Record Linkage employs similarity-preserving encoding on quasi-identifiers, allowing linkage to be performed on encoded representations rather than raw data. This approach protects identities while still facilitating record matching based on encoded similarities and probabilistic approaches. But this also causes the security of Privacy-Preserving Record Linkage schemes to be dependent on the encoding techniques used [SAH24; VCRS20].

Broadly, Privacy-Preserving Record Linkage methods fall into two categories, perturbation based techniques and Secure Multi-Party Computation based techniques. Secure Multi-Party Computation-based techniques provide strong security guarantees and high accuracy but suffer from computational and communication overheads. Conversely, perturbation-based techniques

balance linkage quality, scalability, and privacy protection, making them more practical for real-world applications [VCRS20].

As seen in Figure 2.1 a typical perturbation-based Privacy-Preserving Record Linkage system involves three main parties working together in four steps to enable the linkage while maintaining privacy. First data owners, who are responsible for maintaining their respective databases, $D$ and $D'$, encode the quasi-identifiers by agreeing on an encoding scheme with the corresponding parameters before then as a second step sharing their respective data sets with the linkage unit. The linkage unit, a trusted entity, performs the actual record linkage using only the encoded representations, without access to the original identifiers. Once the linkage is completed, the linkage unit assigns unique pseudonyms to the successfully linked records and returns the pseudonymized dataset as a third step to the data owners. The data owners then replace the linkage data with these pseudonyms before sending the dataset in the fourth and final step to the receipient requesting the data. In the case of a research project this could be an data analyst. The data analyst can merge the records based on the identifiers and proceed with further research and analysis, all while minimizing the risk of re-identification and protecting individual privacy [SAH24].

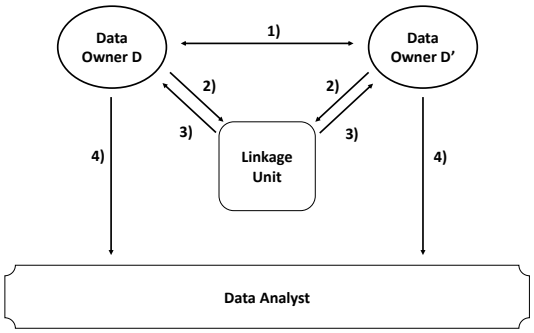


**Figure 2.1:** Overview of the Privacy-Preserving Record Linkage process.

Based on this scheme, each database record can therefore during the linkage process be represented as $r = (\lambda, \sigma)$, where $\lambda$ denotes the linkage data which are encoded quasi-identifiers such as names and birthdates and $\sigma$ refers to the remaining microdata like in a health care scenario patient information [SAH24].

Linkage in Privacy-Preserving Record Linkage is performed probabilistically, meaning that two records, $r$ and $r'$, are considered linked if their similarity score on $sim(\lambda_r, \lambda_{r'})$ exceeds a predefined threshold. The choice of threshold plays an important role in balancing the quality of the linkage. Lower thresholds tolerate more variation and matches between records but increase the likelihood of false positives, while higher thresholds reduce false positives but may miss legitimate matches. Thus, selecting the optimal threshold is a trade-off that requires careful consideration of the specific goals and constraints of the linkage process [SAH24].

A robust Privacy-Preserving Record Linkage scheme must satisfy several important criteria to ensure effective and secure linkage. First, a similarity function, $sim(\lambda_r, \lambda_{r'})$, must exist to determine if two records belong to the same entity based on a predefined threshold. Second, an encoding scheme $enc(\lambda)$ must be applied to $\lambda$ in such a way that the linkage unit cannot reconstruct the original data. Finally, a function $sim(enc(\lambda_r), enc(\lambda_{r'}))$ must be available that

allows similarity computations on encoded data. These requirements ensure both the privacy and the effectiveness of the record linkage process [SAH24].

One common approach for measuring similarity for two quasi identifiers and enabling probabilistic matching on quasi identifieres is using n-grams. Here, string values are divided into overlapping substrings of length $n$ using a sliding window approach [SAH24]. For example, for the string "encoding" with $n = 2$, the n-grams are:

$$\{\text{en}, \text{nc}, \text{co}, \text{od}, \text{di}, \text{in}, \text{ng}\}$$

The similarity of two sets of n-grams can then be computed using metrics such as the Dice Coefficient where X and Y denotes two sets which are in the case of Privacy-Preserving Record Linkage the n-grams for two different pseudo identifiers [SAH24].

$$Dice(X, Y) = \frac{2 \cdot |X \cap Y|}{|X| + |Y|}$$

The Jaccard Similarity can be used in a similar way [SAH24].

$$Jaccard(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

Privacy-Preserving Record Linkage has been successfully applied in various domains, demonstrating its practical importance in securely linking records across institutions while trying to preserve privacy. Notable applications include the Social Investment Data Resources, the Lumos Initiative, the Swiss National Cohort, and the Gemeinsamer Bundesausschuss. These implementations highlight the versatility of Privacy-Preserving Record Linkage in diverse contexts, where privacy protection is essential while enabling effective data linkage for research and analysis [SAH24].

## 2.2 Key Encoding Techniques

In the context of Privacy-Preserving Record Linkage, three primary encoding techniques have emerged: Bloom Filter, Tabulation MinHash and Two-Step Hash. These encoding methods are essential for transforming sets of quasi-identifiers into representations that preserve similarity information while trying to maintain privacy [SAH24; SBR09; VCRS20].

Privacy-Preserving Record Linkage typically involves encoding sets of quasi-identifiers before linkage. A set in this context is generally a collection of n-grams, which are substrings of length $n$ extracted from one or multiple attributes using a sliding window approach. Since input data varies in length, all encoding techniques in Privacy-Preserving Record Linkage must take arbitrarily long inputs (sets of n-grams) and produce encoded outputs. This ensures that similarity computations can be efficiently performed on encoded data by the linkage unit without accessing the raw identifiers [SAH24; VCRS20].

Before linkage, data owners must agree on the encoding scheme and share necessary cryptographic secrets to facilitate secure comparison. Among the available techniques, Bloom Filters are the most widely used approach for record linkage [SAH24].

### 2.2.1 Bloom Filter

Bloom Filters were originally developed for efficient membership testing in set structures without requiring direct access to the sets themselves [Blo70]. Due to their ability to efficiently compute set similarities in a privacy-preserving manner, they have been widely adopted in Privacy-Preserving Record Linkage applications [SAH24; SBR09; VCRS20].

A Bloom Filter $b \in \{0,1\}^l$ is a bit vector of length $l$. It uses $k \geq 1$ independent hash functions $H = \{h_1, h_2, ..., h_k\}$, where each function maps an arbitrary input to a position in the filter [SAH24; SBR09]:

$$h_i : \{0,1\}^* \rightarrow \{1, ..., l\}, \quad \forall i \in \{1, ..., k\} \tag{2.1}$$

Initially, the Bloom Filter is set to all zeros. Each element $s \in S$ is hashed with every function $h_i$, and the corresponding bit positions in the filter are set to 1 [SAH24; SBR09]:

$$\forall s \in S, \forall h_i \in H, \quad b[h_i(s)] = 1 \tag{2.2}$$

An Example for this can be seen in Figure 2.2 where the set of 2-grams for the word "encoding" is encoded using a Bloom Filter with $k = 2$ hash functions. As can be seen, the 2-grams are hashed using the two hash functions and the corresponding bits are set to 1 in the Bloom Filter.

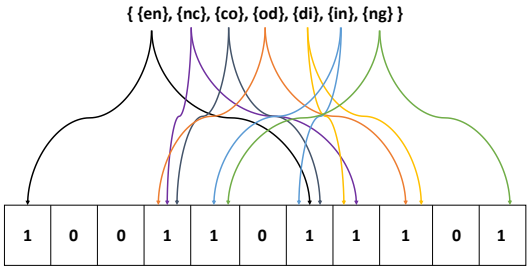


**Figure 2.2:** Example Bloom Filter for $k = 2$ hash functions on the set of 2-grams for "encoding".

Since Bloom Filters are binary vectors, the similarity between two Bloom Filters, $b_1$ and $b_2$, is computed based on the overlapping 1-bits based on their position. The Dice Coefficient is commonly used for this purpose [SAH24], providing a measure of similarity by comparing the overlap of 1-bits in the two binary vectors. Therefore encodings using Bloom Filter encoding allows for efficient computation of similarity between two sets which is benefical for Privacy-Preserving Record Linkage systems.

$$\text{Dice}(b_1, b_2) = \frac{2 \cdot |b_1 \cap b_2|}{|b_1| + |b_2|} \tag{2.3}$$

Because the sets in a Privacy-Preserving Record Linkage systems consists of n-grams, a deterministic relationship between the n-grams present in the quasi-identifier $\lambda$ and the set bits in the Bloom Filter is created. However, due to the finite length of Bloom Filters, collisions

occur where different n-grams map to the same bit position (see Figure 2.2). While this can cause incorrect linkages, it also enhances privacy by distorting frequency distributions [SAH24; VCRS20].

Three primary approaches exist for applying Bloom Filters to sensitive data. The first approach, Attribute-Level Bloom Filters, encodes each attribute, such as first name or last name, into a separate Bloom Filter, enabling multiple similarity computations. However, Attribute-Level Bloom Filters are more vulnerable to frequency-based privacy attacks as they lower the collisions which would occur using only one Bloom Filter. The second approach, Cryptographic Long-Term Key Encoding, merges multiple attributes into a single Bloom Filter, reducing vulnerability to frequency attacks but remaining susceptible to pattern-mining-based attacks. Finally, Record-Level Bloom Filters employ a weighted bit sampling technique to minimize frequency information, enhancing privacy protection while maintaining high linkage quality. Each of these approaches balances privacy concerns with the need for accurate and effective record linkage [VCRS20].

Several privacy-enhancing methods have been proposed to mitigate frequency attacks in Bloom Filters. These techniques introduce a trade-off between privacy and linkage quality. One such method is balancing, which ensures an equal number of 1-bits across Bloom Filters, thereby reducing the likelihood of frequency-based attacks. Another approach is salting, which randomizes bit positions to prevent direct inference from the encoded quasi-identifiers. Additionally, XOR folding is used to reduce the Bloom Filter length while maintaining the bit-wise dependencies necessary for effective linkage. These methods aim to strengthen privacy while retaining the accuracy of the linkage process [SAH24; VCRS20].

A major improvement was introduced by Armknecht et al. [AHS23], who proposed a diffusion layer for Bloom Filter encodings. This method generates Encoded Linkage Data, where each bit is computed as the XOR sum of multiple Bloom Filter bits. The indices for XOR computations are randomly chosen and secretly shared among data owners [AHS23].

By applying diffusion, the deterministic relationship between 1-bits in the Encoded Linkage Data and the original n-grams is broken, improving privacy while still enabling approximate matching [AHS23].

Bloom Filters remain a core technique in Privacy-Preserving Record Linkage due to their efficiency and scalability. However, their vulnerability to frequency attacks has led to improvements such as Record-Level Bloom Filters and diffusion layers, which enhance privacy at the cost of increased computational complexity [AHS23; SAH24; VCRS20].

### 2.2.2 Tabulation MinHash

Tabulation MinHash is a variation of MinHash initially introduced for efficient estimation of set similarities and later adapted for privacy-preserving probabilistic record linkage. MinHash itself was first proposed in the context of document resemblance and containment estimation. Tabulation MinHash extends MinHash by employing tabulation-based hashing, which enhances its security compared to Bloom Filters [Bro97; VCRS20].

MinHash aims to approximate the Jaccard similarity between two sets, $S$ and $S'$. The fundamental idea is to represent both sets as sequences of randomly ordered elements and apply multiple rounds of random permutations $\pi$ to shuffle them. After each round, the first elements of both sequences are compared. The larger the intersection between $S$ and $S'$, the higher the probability that the first elements will match. An example for this can be seen in Figure 2.3 where the sets are permutated twice and the first element is compared for each

new set. The final Jaccard similarity estimate is computed based on the number of collisions achieved during permutation, in the example is one collision for two permutations [Bro97; SAH24; VCRS20].

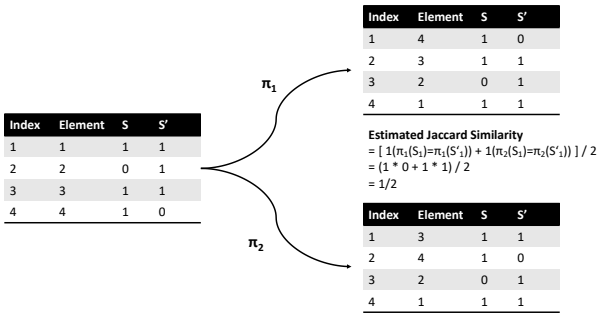


**Figure 2.3:** Example computing approximate Jaccard similarity using MinHash with $\pi = 2$ permutations.

Instead of explicitly computing these permutations, MinHash simulates them by applying a suitable hash function to the elements of a set and selecting the smallest hash value as the representative signature. This is equivalent to sorting set elements by their hash values and returning the first element [SAH24].

Tabulation-based hashing is a technique used in Tabulation MinHash that provides efficient and high-quality hash functions by leveraging precomputed lookup tables. This method operates as follows [VCRS20]:

The process begins with the **initialization** of $l$ sets of lookup tables, each containing $c$ tables. Each table holds randomly generated bit strings for keys of length $k$, with a key space of $2^k$. During the **hashing process**, each element in $S$ is hashed using a one-way hash function, producing a fixed-length binary value. This binary value is then split into $c$ sub-keys, each of length $k$. Each sub-key is used as an index to retrieve a random bit string from the corresponding lookup table, and the retrieved $c$ bit strings are XORed together to produce a single output bit string. In the **MinHash signature generation** phase, this process is repeated for each of the $l$ lookup table sets, and the minimum value among all generated bit strings is selected as the MinHash signature [SAH24; VCRS20].

An example for this can be seen in Figure 2.4 where the first element of the set is hashed using the first lookup table and the fourth hash function. The key is split into $c = 4$ sub-keys of length $k = 2$ and used to retrieve the corresponding bit strings from the lookup table. The retrieved bit strings are then XORed together to produce the final value for the first element.

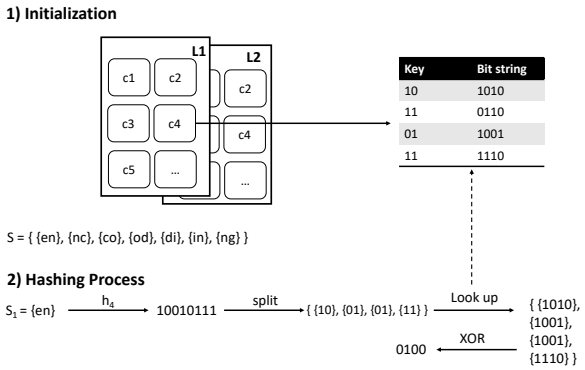


**Figure 2.4:** Simplified Tabulation MinHash hashing step for the first lookup table, fourth hash function on the first set element.

To further enhance privacy, Tabulation MinHash employs a 1-bit hashing mechanism, where only the least significant bit of each MinHash signature is retained. These $l$ bits are then concatenated to form the final bit array used as an encoded representation [SAH24].

The main advantage of Tabulation MinHash over Bloom Filters is its improved resistance to frequency-based attacks due to the complexity introduced by tabulation-based hashing. However, this security enhancement comes at a cost. It leads to higher computational overhead, as the need to generate and access multiple lookup tables increases processing time. Additionally, there is increased memory consumption because storing large precomputed tables requires additional space. These trade-offs must be considered when choosing between Tabulation MinHash and other privacy-preserving techniques [SAH24; VCRS20]. Despite these trade-offs, Tabulation MinHash remains an attractive alternative for Privacy-Preserving Record Linkage due to its robustness against adversarial attacks.

Similar to Bloom Filters, Tabulation MinHash encodes each record as a bit vector of length $l$. Given two Tabulation MinHash-encoded bit vectors, their similarity can be estimated using a modified Jaccard coefficient, adapted to account for artificial bit collisions caused by truncation to the least significant bit. The Jaccard coefficient can also be converted into the Dice coefficient for improved comparability with Bloom Filter-based methods [SAH24; VCRS20].

Overall, Tabulation MinHash provides a more secure encoding alternative to Bloom Filters in Privacy-Preserving Record Linkage, though at the expense of increased computational and memory requirements [SAH24; VCRS20].

### 2.2.3 Two-Step Hash

Two-Step Hash is the most recent encoding scheme proposed for Privacy-Preserving Record Linkage, introduced in 2020 [RCS20]. Two-Step Hash was designed to address both the privacy vulnerabilities of Bloom Filters and the computational complexity of Tabulation MinHash while maintaining accuracy in similarity calculations. Similar to other encoding techniques, Two-Step Hash requires the input to be split into a set of n-grams $S$ prior to encoding [RCS20].

As a result of Two-Step Hash encoding, each record from a sensitive database is represented by a set of integers, which can be directly used to compute Jaccard similarity. Two-Step Hash employs two distinct hashing steps. In the first hashing step, the input set is converted into a bit matrix representation. In the second hashing step, the bit matrix columns are

mapped into integers, enabling efficient comparison. This two-step process allows Two-Step Hash to represent sensitive data in a way that facilitates effective similarity computation while preserving privacy [SAH24]. These steps provide accurate Jaccard similarity calculations while improving privacy protection compared to traditional Bloom Filter-based encodings [RCS20].

In the first step of the Two-Step Hash process, elements of the n-gram set $S$ are hashed into $k$ independent Bloom Filters $b_i$ of length $l$, meaning each hash function results in a corresponding bit vector. This generates a $k \times l$ matrix, where each row corresponds to a Bloom Filter created using a unique hash function, and each column represents the bitwise state across all Bloom Filters for a given position. In the second step, after constructing the bit matrix, Two-Step Hash computes column-wise hashes to convert the bit vectors into integer representations. Each column vector is treated as an input for a hash function, and all-zero columns are skipped to prevent distortion in similarity calculations since they do not encode any n-grams. To enhance security and avoid hash collisions between columns with identical bit patterns, a salt value and the column index are concatenated before hashing [RCS20].

The final integer representation for each column $i$ for $1 \leq i \leq l$. is computed as [RCS20; SAH24]:

$$H(salt, i, b_{1i}, b_{2i}, ..., b_{ki}) \tag{2.4}$$

The output of the second hashing step is a set of integers, allowing similarity computations using the Dice coefficient rather than directly computing bitwise similarity. Since the encoded data consists of sets, Jaccard similarity can also be computed similarly to MinHash-based encodings [RCS20].

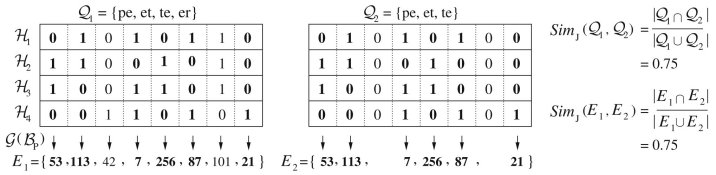


**Figure 2.5:** Two-Step Hash example for two input values "peter" and "pete" [RCS20].

An illustrative example is provided in Figure 2.5, where the set of 2-grams for the words "*peter*" and "*pete*" is encoded using Two-Step Hash with $k = 4$ hash functions and a Bloom Filter length of $l = 8$. In the first hashing step, each 2-gram is processed using the $k$ hash functions, resulting in a $4 \times 8$ bit matrix. Each row of this matrix corresponds to a Bloom Filter generated with a distinct hash function, encoding the presence of 2-grams across the $l$ bit positions.

In the second hashing step, the columns of the bit matrix are transformed into integer values. This is achieved by applying an additional hash function that incorporates both a salt value and the column index, effectively compressing the binary representation into a fixed-length integer vector. The final output is a set of integers representing the encoded word, which can

subsequently be used to compute similarity scores between different words [RCS20].

To improve efficiency, Two-Step Hash can be implemented using a Pseudo-Random Number Generator instead of cryptographic hash functions. The Pseudo-Random Number Generator is seeded with the value to be hashed before generating random numbers, ensuring that the sequence of generated values depends deterministically on the input [RCS20].

By combining efficient bit vector representations with integer-based similarity computations, Two-Step Hash offers a balance between privacy, security, and computational efficiency, making it a promising alternative to existing Privacy-Preserving Record Linkage encoding schemes [RCS20; VCRS20].

## 2.3 Graph Matching Attack

Graph Matching Attacks were introduced by Vidanage et al. [VCRS20] and represent the most significant threat to Privacy-Preserving Record Linkage due to their universal applicability. Unlike traditional cryptanalytic attacks, Graph Matching Attacks exploit the fundamental properties of non-interactive Privacy-Preserving Record Linkage to compromise the security of all schemes relying on similarity-preserving encoding [SAH24].

Non-interactive Privacy-Preserving Record Linkage refers to linkage schemes where data owners independently encode their data and share it with a linkage unit. The linkage unit then performs record matching solely based on the encoded data, without requiring further interaction with the data owners during the linkage process. This approach minimizes communication overhead and computational complexity, as no iterative exchanges between parties are necessary. In contrast, interactive PPRL methods involve multiple rounds of communication between data owners and the linkage unit to refine matching results or improve accuracy [KKM+14].

In Privacy-Preserving Record Linkage, encoded records are linked based on similarity computations. Since these similarities serve as identifiers, an attacker with access to an encoded dataset can leverage an auxiliary plaintext dataset to re-identify individuals. The latest version of the Graph Matching Attack developed by Schaefer et al. [SAH24] overcomes the limitations of the original attack by Vidanage et al. [VCRS20] and enhances success rate and robustness, even under limited knowledge scenarios [SAH24].

In the context of a Graph Matching Attack on a Privacy-Preserving Record Linkage system, the attacker is modeled as the linkage unit and is assumed to have minimal prior knowledge. The attacker does not know any encoding secrets, seeds, or salts used in the system to protect the data. The only information available to the attacker is that which is inevitably known to the linkage unit during the linkage process. This assumption ensures that the attacker can only exploit data accessible through normal system operations, adhering to Kerckhoffs's principle [SAH24].

Since the attack does not depend on specific encoding parameters or attribute frequency distributions, it is universally applicable as long as pairwise similarities of encoded data are available [SAH24].

The first step of the attack involves constructing similarity graphs for both the encoded dataset ($D_{enc}$) and the plaintext dataset ($D_{plain}$). In these graphs, each node represents an individual record, while edges between nodes are assigned weights based on pairwise similarity computations. To ensure computational efficiency and focus only on meaningful connections, edges with similarity scores below a predefined threshold are omitted, reducing noise and

improving the accuracy of the attack [SAH24].

Since certain encoding properties, such as Bloom Filter length ($l$), are inevitably known to the linkage unit, $D_{plain}$ can be transformed analogously to $D_{enc}$ for effective comparison. Importantly, this step does not require knowledge of shared secrets, as the primary objective is to replicate the effect of encoding on similarity [SAH24].

To quantify the structural similarity between nodes in $G_{plain}$ and $G_{enc}$, node embeddings are computed to transform the graph structure into a numerical representation. This process begins with graph embedding using the Node2Vec algorithm, which applies a Word2Vec-like approach to learn vector representations of nodes. During this process, nodes undergo multiple random walks, where each walk simulates a sequence of transitions between connected nodes. These sequences are then treated as sentences, allowing the model to learn embeddings that capture the local and global structure of the graph. The behavior of these random walks is controlled by two hyperparameters: $p$, which determines the likelihood of returning to a previously visited node, and $q$, which influences the tendency to explore new regions of the graph. The result is an embedding matrix where each row represents a node as a vector in Euclidean space [SAH24].

Once embeddings are generated, they must be aligned to allow meaningful comparison between the two graphs. Due to the randomness inherent in embedding generation, direct comparison is not possible. Instead, an iterative approach is used to solve two subproblems: first, an optimal linear transformation is determined using Procrustes Analysis to align the embeddings, and second, node correspondences are established via the Sinkhorn Algorithm, which minimizes the Wasserstein distance between the distributions of embeddings in both graphs. To achieve an effective alignment, an unsupervised stochastic optimization scheme alternates between these two steps over $n$ epochs, gradually refining the transformation and correspondences until convergence [SAH24].

Once the embeddings from the plaintext and encoded datasets are aligned, the reidentification process can begin. Each embedding in the transformed plaintext space is compared to its counterparts in the encoded space, with similarity measured using cosine similarity. This metric quantifies how closely two embeddings align in the high-dimensional space, enabling the attacker to identify records in the encoded dataset that most closely resemble those in the plaintext dataset [SAH24].

The final step involves constructing a bipartite graph, where nodes from the plaintext and encoded datasets are linked based on their similarity scores. To determine the optimal mapping, the Jonker-Volgenant algorithm is applied, ensuring that each node in the smaller dataset is uniquely matched to a corresponding node in the larger dataset. This algorithm maximizes the total similarity across all matched pairs, effectively revealing the identities of individuals within the encoded dataset [SAH24].

An example of this process is illustrated in Figures 2.6 and 2.7, which provide a high-level overview of the Graph Matching Attack attack applied to a Privacy-Preserving Record Linkage approach using Bloom Filter encoding. Initially, the two data owners agree on an encoding scheme and encode their respective datasets, for this example using Bloom Filter. These encoded datasets are then send to the linkage unit. The attack begins at this point, leveraging information inherently available to the linkage unit. Specifically, the linkage unit constructs similarity graphs for both the encoded datasets and an auxiliary plaintext dataset. By embedding these similarity graphs into a vector space and aligning the embeddings, the attacker can identify re-identifications by comparing embeddings and constructing a bipartite graph that matches entries from the encoded dataset to those in the plaintext auxiliary dataset

[SAH24].

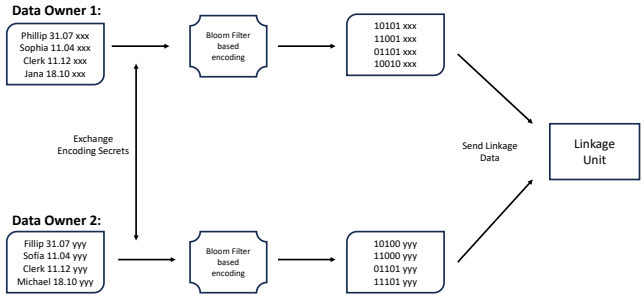


**Figure 2.6:** High-level overview of the Graph Matching Attack attack process. Two data owners encode their datasets and send them to the linkage unit.

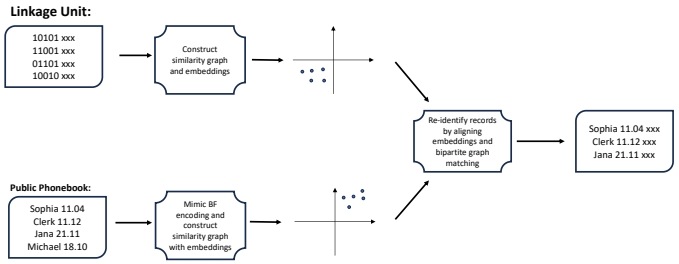


**Figure 2.7:** High-level overview of the Graph Matching Attack attack process. The linkage unit mimics the Bloom Filter encoding for the public dataset and creates for both datasets similarity graphs, embeddings and aligns them to perform bipartite matching.

The improved Graph Matching Attack approach by Schaefer et al. [SAH24] achieves near-perfect re-identification rates when dataset overlap is 100%. Even for low-overlap scenarios (e.g., 5%), success rates reach 99.9% for Two-Step Hash [SAH24]. The only encoding scheme resistant to Graph Matching Attacks is Bloom Filters with diffusion layers, which disrupts similarity preservation for sufficiently high diffusion values [SAH24].

## 2.4 Artificial Neural Network

Artificial Neural Networks are a class of Machine Learning models inspired by the structure and function of biological neural systems. They consist of interconnected layers of artificial neurons that process input data and extract meaningful patterns through iterative learning. Artificial

Neural Networks have been widely applied in various fields, including image recognition, natural language processing, and classification tasks. Neural networks are particularly effective for complex tasks because they automatically identify and refine patterns in data through multiple layers of processing, removing the need for manual feature engineering, which can be difficult and time-consuming [DKK+12].

The structure of an Artificial Neural Network consists of multiple layers as can be seen in Figure 2.8, each serving a distinct role in processing and transforming input data. The input layer is the first stage of the network, responsible for receiving raw data and forwarding it to subsequent layers. The number of neurons in this layer corresponds directly to the number of input features, ensuring that all relevant information is passed through the network [DKK+12; SMNP24].

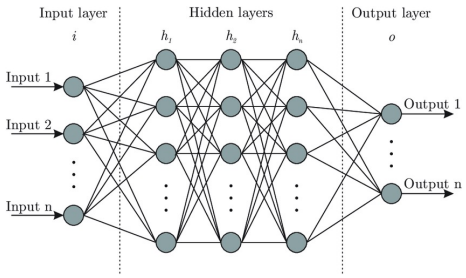


**Figure 2.8:** Artificial Neural Network consisting of multiple input neurons (input layer), hidden layers and output neurons (output layer) [BGF17].

Following the input layer are the hidden layers, which perform feature extraction and transformation. Each neuron in a layer applies a weighted sum operation to its inputs, followed by an activation function that introduces non-linearity, enabling the network to learn complex patterns in the data. Common activation functions include the Rectified Linear Unit (ReLU), Sigmoid, Leaky ReLU, and Exponential Linear Unit (ELU), each offering advantages depending on the specific task [RN16; SSA17].

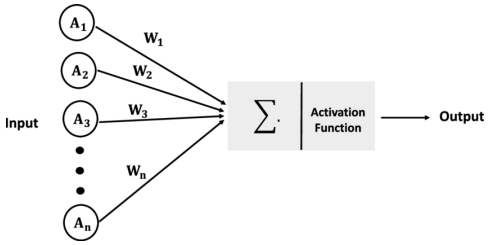


**Figure 2.9:** Sketch of an single artifical neuron in an Artificial Neural Network [GSB+21].

An example of this process is illustrated in Figure 2.9, which depicts a single neuron in an Artificial Neural Network. The neuron takes multiple inputs, multiplies them by corresponding weights, sums the weighted inputs along with a bias term, and applies an activation function to compute the final output. Mathematically, this operation is represented as

$$a = \sigma(z) = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2.5}$$

where $a$ is the neuron's output, $\sigma$ is the activation function, $z$ is the weighted sum, $w_i$ are the weights, $x_i$ are the inputs, and $b$ is the bias term [GSB+21]. The depth and size of the hidden layers determine the network's capacity to model relationships, making them an important component of deep learning architectures [DKK+12; SMNP24].

Finally, the output layer generates the final predictions based on the processed information. The number of neurons in this layer depends on the nature of the task, whether it is a classification problem, where each neuron represents a class, or a regression task, where a single neuron outputs a continuous value [DKK+12].

Training an Artificial Neural Network involves iteratively adjusting its parameters, using a labeled dataset to minimize prediction errors. The process begins with forward propagation, where input data flows through the network, passing through multiple layers until it reaches the output layer, generating a prediction. This prediction is then compared to the actual target value, and the discrepancy between the two is quantified using a loss function, which measures the model's performance [RN16].

To improve accuracy, the network undergoes backward propagation (backpropagation), where the gradient of the loss with respect to each weight is computed using the chain rule of differentiation. These gradients indicate how each parameter should be adjusted to reduce the overall error [RN16].

An optimizer, such as Stochastic Gradient Descent (SGD) or Adam, updates the weights accordingly by taking small steps in the direction that minimizes the loss. The training process is repeated over multiple epochs, where the entire dataset is processed multiple times. To enhance efficiency, the data is often divided into batches, allowing the model to update its weights incrementally rather than processing the entire dataset at once. Over time, this iterative optimization process enables the network to learn meaningful patterns and improve its predictive performance. Artificial Neural Networks can be applied to different classification tasks, depending on whether a data instance belongs to a single category or multiple categories simultaneously [RN16].

In traditional single-label or binary classification problems, each instance is assigned to one and only one category from a predefined set of classes. To achieve this, the network's output layer typically uses a Softmax activation function, which converts the raw output scores into a probability distribution over all possible classes. The model can be trained using the Cross-Entropy Loss function, which penalizes incorrect classifications by measuring the difference between the predicted probability distribution and the actual class label [HCR+16; RN16].

In contrast, multi-label classification allows an instance to belong to multiple categories (labels) at the same time. Instead of a single categorical output, the network produces independent predictions for each possible label. The output layer can then as an example apply sigmoid activations for each label, transforming the raw scores into independent probabilities indicating the presence or absence of each class. Since each label is treated as a separate binary classification problem, Binary Cross-Entropy (BCE) Loss is commonly used to optimize the model, ensuring accurate predictions across multiple labels [HCR+16; RN16].

Different Artificial Neural Network architectures have been developed to address various problem domains, each optimized for specific types of data and tasks. Feedforward Artificial Neural Networks (FNNs) represent the simplest architecture, where data flows in one direction

from the input layer to the output layer without forming cycles. These networks are widely used for basic classification and regression tasks but may struggle with complex patterns that require spatial or sequential dependencies [GB10; RN16].

For tasks involving image processing, Convolutional Artificial Neural Networks (CNNs) are commonly used. CNNs employ convolutional layers that apply filters to input images, allowing the network to capture complex patterns such as edges, textures, and shapes. This makes them highly effective for applications like object recognition and medical imaging [ON15; SMNP24].

When dealing with sequential data, Recurrent Artificial Neural Networks (RNNs) and their advanced variant, Long Short-Term Memory (LSTM) Networks, are particularly useful. These architectures introduce recurrent connections, enabling them to maintain memory of previous inputs and recognize patterns over time. This makes them well-suited for natural language processing, speech recognition, and time series forecasting [MJ+01].

Despite their success, Artificial Neural Networks present several challenges that must be managed to ensure robust and efficient learning. One of the most common issues is overfitting, where a model becomes too specialized in learning patterns from the training data, capturing possible noise rather than generalizable features. This leads to poor performance on unseen data. Techniques such as dropout regularization, L2 weight decay, and early stopping are commonly used to mitigate overfitting and improve generalization [Fou; GB10].

Dropout regularization is a technique used to prevent overfitting by randomly setting a fraction of neurons to zero during training. This forces the network to learn more robust features by preventing it to rely too heavily on a single neuron to perform well. L2 weight decay is another regularization method that penalizes large weights by adding a regularization term to the loss function. It tries to prevent the network from applying too much importance to a single feature, encouraging it to learn more generalizable patterns. Early stopping is a simple yet effective technique that stops training when the model's performance on a validation set starts to degrade below a certain threshold, preventing overfitting [Fou; GB10].

Another fundamental challenge is the vanishing and exploding gradient problem, which occurs in deep networks during backpropagation. When gradients become too small (vanishing), weight updates diminish, leading to slow or stalled learning. This can happen because gradients are the product of multiple derivatives, which can cause them to shrink exponentially as they propagate through the network. This can happen especially using activation functions that saturate for extreme values. Conversely, when gradients grow too large (exploding), unstable updates cause erratic training behavior. Solutions such as batch normalization, gradient clipping, and advanced activation functions like Leaky ReLU help address these issues [Fou; GB10].

Batch normalization stabilizes training by normalizing the inputs to each layer, ensuring that activation values remain within a consistent range. Gradient clipping addresses the problem of exploding gradients by imposing a threshold on gradient values, thereby maintaining training stability. Leaky ReLU, an activation function, mitigates the vanishing gradient problem by allowing a small, non-zero gradient for negative inputs, enabling continued learning even with extreme input values [Fou; GB10].

The computational complexity of deep learning models is another major concern, as large-scale Artificial Neural Networks require extensive memory and processing power. Training deep networks on large datasets can be prohibitively slow on CPUs, necessitating the use of GPUs or specialized hardware like TPUs (Tensor Processing Units) to accelerate training. Efficient data-loading techniques and mixed-precision training can further optimize computational efficiency. Mixed-precision training is a technique that partly uses lower-precision floating-point numbers

to reduce memory usage and speed up computations, while still maintaining model accuracy [Fou].

Lastly, hyperparameter tuning plays an important role in model performance. Selecting the right learning rate, batch size, number of layers, and optimization algorithm requires experimentation and fine-tuning. Automated methods such as grid search, random search, and Bayesian optimization can assist in finding optimal configurations, but these processes are computationally expensive. Addressing these challenges effectively is important for developing high-performing Artificial Neural Networks that generalize well across different datasets and tasks [Fou].

# 3 Methodology

The Dataset Extension Attack is a novel attack method that extends the capabilities of Graph Matching Attacks by moving beyond the intersection of datasets to re-identify individuals who were previously unmapped. This chapter outlines the methodology behind the Dataset Extension Attack, including modifications to the Graph Matching Attack, the design and implementation of the Dataset Extension Attack itself, and the use of Artificial Neural Networks to enable probabilistic reconstruction of Personally Identifiable Information from encoded data.

The Dataset Extension Attack builds upon the Graph Matching Attack by using its re-identification results as a foundation for further inference. While the Graph Matching Attack re-identifies only those records that exist in both the attacker's dataset and the encoded target dataset, it can leave a substantial portion of records unmapped. The goal of the Dataset Extension Attack is to extend this re-identification process by applying a Machine Learning–based approach to infer the missing Personally Identifiable Information of the remaining records using a trained Artificial Neural Network.

To achieve this, the Dataset Extension Attack follows a structured pipeline comprising six key steps, as illustrated in Figure 3.1. The first step involves executing the Graph Matching Attack and extracting its results in a predefined format to serve as training data. This dataset includes the re-identified individuals, their corresponding encoded representations, and the plaintext information that was successfully linked. In addition, the Graph Matching Attack results contain the non-re-identified individuals, who are represented solely by their encoded Personally Identifiable Information and associated plaintext values.

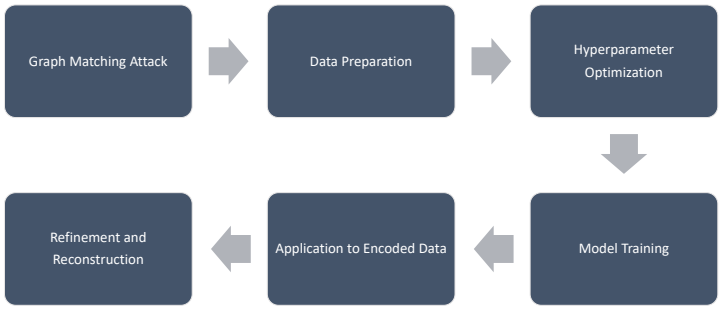


**Figure 3.1:** Overview of the Dataset Extension Attack attack pipeline.

Once the data is extracted, it undergoes a transformation process to prepare it for Artificial Neural Network training. This data preparation step involves constructing specialized datasets that convert the encoded representations and their corresponding labels, plaintext n-grams, into tensor-based formats suitable for processing by deep learning models. The resulting datasets are subsequently split into training, validation, and test subsets, and corresponding data loaders

are created to facilitate efficient mini-batch processing during model training.

With the data pipeline established, hyperparameter optimization is performed to determine the most effective model configuration. This process systematically explores combinations of hyperparameters, such as the number of hidden layers, hidden layer size, activation functions, optimizers, and learning rate schedulers, to identify those that yield the best performance. Hyperparameter tuning is essential because different encoding schemes with different parameters necessitate tailored model configurations to effectively capture the underlying patterns in the data. The selected hyperparameters per trial are used to define the Artificial Neural Network architecture, which is trained to learn the mapping between encoded representations and their corresponding plaintext n-grams.

The architecture of the Artificial Neural Network is tailored to the specific characteristics of the encoding schemes used in the Privacy-Preserving Record Linkage scheme. The input layer size is determined by the dimensionality of the encoded representation, while the output layer size corresponds to the size of a predefined n-gram dictionary.

Once the best hyperparameter configuration is identified, the Artificial Neural Network is trained using the re-identified individuals as labeled data. Training proceeds over multiple epochs, during which the model iteratively processes the training dataset, computes the loss, and updates its parameters via backpropagation. Performance is continuously monitored on the validation set to track generalization and prevent overfitting.

Once the Artificial Neural Network is trained, it can be applied to the set of non-re-identified individuals, i.e., records that remained unmapped after the Graph Matching Attack. This dataset serves as the test set during the experimentation phase to evaluate the performance of the attack.

The model outputs a probability distribution over an index that maps to a dictionary of n-grams, indicating the likelihood of each n-gram being present in the corresponding plaintext. To refine these predictions, a thresholding mechanism is applied to filter out low-confidence outputs and retain only the most probable n-grams. These predicted n-grams are then reconstructed into potential Personally Identifiable Information, constituting the final step of the Dataset Extension Attack process.

This methodological approach represents an advancement in attacking Privacy-Preserving Record Linkage systems. By leveraging deep learning techniques, the Dataset Extension Attack enables an attacker to infer sensitive personal information beyond the scope of traditional Graph Matching Attack approaches. The following sections provide a detailed discussion of each component, including the design choices, implementation details, and challenges encountered during development.

## 3.1 Problem Definition

The primary challenge that the Dataset Extension Attack seeks to address is the limited scope of re-identifications achieved by the Graph Matching Attack. While the Graph Matching Attack effectively links records by exploiting structural relationships within encoded datasets, its success is inherently restricted to individuals who are present in both the plaintext and the encoded datasets and can be matched based on graph similarity. However, in real-world scenarios, there may exist additional re-identification potential beyond these direct matches.

One possible way to extend re-identifications is to rerun the Graph Matching Attack iteratively, incorporating additional publicly available data to gradually refine the matching process.

However, this approach is dependent on the availability and quality of external data sources, which may not always be feasible. Instead, the Dataset Extension Attack introduces a novel strategy that aims to reconstruct deterministic relationships between encoded representations and their corresponding plaintext information. This is based on the observation that all encoding schemes used in Privacy-Preserving Record Linkage rely on hash functions or other deterministic mappings.

Hash functions, for example, produce fixed-length outputs from inputs of arbitrary length and are deterministic, meaning the same input will always yield the same output. The Dataset Extension Attack leverages this property by training Artificial Neural Networks to learn statistical relationships between encoded values and the original n-grams of Personally Identifiable Information. The objective is to recover the most probable plaintext representation given an encoded input, effectively framing the attack as a probabilistic, frequency-based inference problem. However, several challenges complicate this task.

The first major challenge is the lack of knowledge about the specific number and type of hash functions used during encoding. As a result, the model must learn patterns in the data without any explicit understanding of the underlying hashing mechanisms. Fortunately, this limitation is partially mitigated by the fact that the Dataset Extension Attack does not rely on a one-to-one mapping between hash outputs and plaintext n-grams, but instead depends on statistical inference across large numbers of training samples.

A more fundamental challenge arises from the collision property of hash functions. Because hash functions map an infinite input space to a finite output space, different inputs may produce identical hash values, making it inherently difficult to perfectly recover the original plaintext. These collisions introduce uncertainty into the re-identification process, preventing the Dataset Extension Attack from achieving perfect reconstruction accuracy. Consequently, the predictions made by the Dataset Extension Attack are probabilistic rather than deterministic. Therefore it can estimate the likelihood of a specific n-gram being present in the original Personally Identifiable Information, but cannot guarantee absolute correctness.

The primary reason the Graph Matching Attack alone is unable to achieve perfect reconstruction is that it relies solely on structural similarities within the dataset, without attempting to infer direct relationships between encoded values and their plaintext equivalents. In contrast, the Dataset Extension Attack enhances the capabilities of the Graph Matching Attack by enabling the reconstruction of individual plaintext components directly from encoded representations, thereby increasing the overall re-identification potential. This novel approach improves the effectiveness of the attack, allowing for the re-identification of individuals who were previously unmatchable using graph-based techniques.

## 3.2 Attacker Model

The attacker in the Dataset Extension Attack scenario is modeled as the linkage unit within a Privacy-Preserving Record Linkage protocol. This aligns with standard threat models in the literature, where the linkage unit is typically assumed to be semi-honest or honest-but-curious executing the prescribed protocol while remaining interested in extracting sensitive information from the encoded data it processes [SAH24].

Following Kerckhoffs's principle, the attacker is assumed to possess full knowledge of the Privacy-Preserving Record Linkage system design, including encoding algorithms (e.g., Bloom Filter construction), parameter settings (e.g., filter length and n-gram size), and record linkage

procedures. However, any encoding specific secrets, such as random seeds and salt values, are assumed to be unknown to the attacker.

In this setting, the attacker is presented with two encoded datasets originating from, for example, two organizations engaging in Privacy-Preserving Record Linkage, and aims to re-identify individuals across them.

The attacker is assumed to operate in an offline setting without time constraints, allowing the use of exhaustive search, large-scale training, and iterative optimization. Given that linkage units are often embedded in national statistical agencies, health departments, or research consortia, it is realistic to assume access to computational resources, including parallel processing and GPU acceleration.

The primary goal of the attacker is to maximize the overall re-identification rate, demonstrating that individuals not re-identified through traditional Graph Matching Attacks can still be decoded using more advanced techniques. To assess the effectiveness of the Dataset Extension Attack, its performance is compared against a baseline strategy that predicts, for each record, the k most frequent n-grams observed in the training data.

This baseline represents a naïve yet plausible strategy. A successful Dataset Extension Attack must outperform this baseline to substantiate its threat to real-world Privacy-Preserving Record Linkage deployments.

## 3.3 Modular Design of the Dataset Extension Attack

The Dataset Extension Attack aims to reconstruct plaintext Personally Identifiable Information from encoded records using Machine Learning techniques. A central challenge in implementing the Dataset Extension Attack lies in the diversity of encoding schemes used to protect sensitive data. As each encoding method transforms plaintext into distinct numerical representations, the Dataset Extension Attack must adapt both the dataset structure and the Artificial Neural Network architecture accordingly.

To address this, the Dataset Extension Attack adopts a modular design: while the overall attack methodology remains consistent, specific implementations are tailored to each encoding scheme. Although the input representation and network architecture vary depending on the encoding, the output format is kept uniform across all models. The attack is framed as a multi-label classification task, where the Artificial Neural Network predicts the likelihood of individual n-grams appearing in the original Personally Identifiable Information. For each encoding scheme, a dedicated dataset structure transforms encoded records into a format suitable for Artificial Neural Network training. Moreover, a custom Artificial Neural Network architecture is employed for each encoding scheme to ensure the model effectively learns the mapping from encoded data to plaintext n-grams.

## 3.4 Step 1: Graph Matching Attack

### 3.4.1 Running the Graph Matching Attack

The Graph Matching Attack constitutes the first step in our Dataset Extension Attack pipeline, establishing a foundation by identifying overlapping individuals between the encoded (target) dataset and the auxiliary (attacker) dataset. It exploits structural similarities between records to perform graph-based matching and re-identification of individuals.

In this phase, we apply the adjusted Graph Matching Attack implementation by Schaefer et al. [SAH24], which builds upon and extends the original approach introduced by Vidanage et al. [VCRS20]. This attack yields a partial mapping between records in the encoded dataset and plaintext identities from the auxiliary dataset. The successfully re-identified individuals represent the known intersection between both datasets and serve as labeled training data for the inference component of the Dataset Extension Attack.

The output of this step comprises two distinct sets of records: (1) re-identified individuals, with known plaintext identities and their corresponding encoded representations, and (2) non-re-identified individuals, whose encoded representations remain unmapped. These unmapped encodings form the target set for the neural network-based reconstruction in the subsequent inference phase.

The effectiveness of the Dataset Extension Attack is directly influenced by the quality of the Graph Matching Attack output. A higher re-identification rate in the Graph Matching Attack provides a larger training set for the Artificial Neural Network, improving its ability to infer plaintext n-grams and reconstruct sensitive information for non-re-identified individuals. Conversely, a low re-identification rate limits the availability of labeled data and diminishes the overall reconstruction capability of the Dataset Extension Attack.

### 3.4.2 Modifications to the Graph Matching Attack

To integrate the Graph Matching Attack as a preprocessing step for the Dataset Extension Attack, modifications were made to the original implementation by Schaefer et al. [SAH24]. While the core algorithm remains unchanged, adjustments were introduced to ensure that the Graph Matching Attack outputs its results in a structured format suitable for training the Artificial Neural Network used in the Dataset Extension Attack.

Originally, the Graph Matching Attack only provided a simple mapping between the IDs of re-identified individuals. However, to enable the Dataset Extension Attack to learn meaningful patterns, access to both the plaintext Personally Identifiable Information and their corresponding encodings is required. Therefore, the Graph Matching Attack was extended to output two datasets in the following format:

- For re-identified individuals: `<Personally Identifiable Information> <encoding> <uid>`
- For non-re-identified individuals: `<encoding> <uid>`

It is important to note that the `uid` is included solely for research and evaluation purposes. It enables researchers to manually track individuals across different processing stages and to assess the performance of the attack. However, in a real-world attack scenario, these `uid`s are neither available nor required. They are entirely excluded from all Dataset Extension Attack training and inference steps, ensuring that the attack methodology remains realistic and practically applicable.

In addition to formatting adjustments, certain components of the Graph Matching Attack were removed to streamline the process and reduce unnecessary complexity. Specifically, encoding schemes other than Two-Step Hash, Tabulation MinHash, and Bloom Filter were excluded, as the Dataset Extension Attack focuses exclusively on these techniques. Other components deemed non-essential, such as graph visualizations and benchmark tests related solely to the

Graph Matching Attack, were also removed. This decision was made because the Graph Matching Attack is not the primary focus of this study; its validity and performance have already been established by prior research. These optimizations resulted in a leaner and more efficient attack pipeline, reducing computational overhead while preserving essential functionality.

With these modifications in place, the starting point for the Dataset Extension Attack is clearly defined. The attack begins with the two structured datasets. By leveraging this structured output, the Dataset Extension Attack can train a Machine Learning model to probabilistically reconstruct missing n-grams from the encoded records of non-re-identified individuals. The following sections detail the implementation of this approach, including dataset preparation, model architecture, and evaluation strategies.

## 3.5 Step 2: Data Representation

For an Artificial Neural Network to operate effectively, the input data must be preprocessed into a format compatible with deep learning models. This preprocessing step applies to both the input, encoded representations of Personally Identifiable Information, and the output, which consists of labels representing the predicted n-grams. This ensures that both re-identified and non-re-identified individuals are structured in a way that enables efficient training and inference.

To facilitate this transformation, custom PyTorch datasets are implemented. These datasets transform the encoded representations into input tensors and represent the associated n-gram labels as binary vectors for multi-label classification, where each position in the label vector indicates the presence or absence of a specific n-gram. This approach enables the model to predict the presence of multiple n-grams per encoded record.

The data representation pipeline is modular and accommodates various encoding schemes, each of which necessitates a tailored preprocessing technique. Depending on the encoding method, such as Bloom Filter, Two-Step Hash, or Tabulation MinHash, different strategies are employed to convert the encoded input into tensors while preserving as much information as possible. This ensures that the input is well-suited to the architecture of the corresponding Artificial Neural Network and that the model can effectively learn the mapping between encoded data and plaintext n-grams.

### 3.5.1 Bloom Filter Encoding

Bloom Filter's are fixed-length binary strings, with their length determined by Alice's chosen encoding parameters. The transformation of a Bloom Filter into a PyTorch tensor is straightforward: each bit in the binary string is directly mapped to a corresponding position in the tensor. This conversion preserves the positions of set bits (i.e., ones), thereby maintaining the structural integrity of the original encoding. The resulting tensor has the same dimensionality as the Bloom Filter, with ones indicating the activated hash positions and zeros elsewhere. This binary representation serves as the input to the Artificial Neural Network, allowing the model to learn patterns based on the bitwise structure of the encoded Personally Identifiable Information.

### 3.5.2 Tabulation MinHash Encoding

Tabulation MinHash, like Bloom Filters, produces fixed-length binary bitstrings, with the specific length determined by the encoding parameters selected by Alice. The transformation into a PyTorch tensor mirrors that of the Bloom Filter: each bit in the Tabulation MinHash string is mapped directly to a corresponding tensor position, preserving the locations of set bits. This direct conversion results in a binary tensor representation that retains the structure of the original Tabulation MinHash encoding. By preserving the positional information of the activated bits, the Artificial Neural Network can effectively learn from the encoded patterns embedded in the Tabulation MinHash representations.

### 3.5.3 Two-Step Hash Encoding

The preprocessing of Two-Step Hash encodings is more complex due to its variable-length representation. Unlike Bloom Filter and Tabulation MinHash, which produce fixed-length binary bitstrings, Two-Step Hash generates a set of integers of arbitrary size. This variability arises because columns containing only zero values are dropped during the Two-Step Hash encoding process.

Since Artificial Neural Networks require fixed-length input vectors, an appropriate transformation is necessary to standardize Two-Step Hash encodings. Simple aggregation techniques, such as averaging, can lead to substantial information loss, particularly problematic in this already knowledge constrained setting. To preserve the richness of the encoded data, an alternative method is employed to convert Two-Step Hash encodings into a tensor compatible format.

To achieve this, all unique integer values from both the re-identified and non-reidentified datasets are collected and stored in a set. This set is then sorted in ascending order and transformed into a dictionary that maps each integer to a unique index. Using this mapping, each Two-Step Hash encoding is converted into a binary vector using a one-hot encoding scheme. For each integer present in the Two-Step Hash encoding, the corresponding index in the binary vector is set to one, while all other positions remain zero.

### 3.5.4 Re-Identified Individuals as Labeled Training Data

To enable supervised learning, re-identified individuals are used as labeled training and validation data. Since their Personally Identifiable Information is known along with their corresponding encoded representation, it is possible to construct datasets where the input consists of transformed encodings (Bloom Filter, Tabulation MinHash, or Two-Step Hash, respectively) into tensors and the output labels consist of the correct n-grams derived from the original Personally Identifiable Information.

To facilitate this process, a predefined dictionary of all possible n-grams is created. This dictionary includes:

- Alphabetical n-grams (e.g., for 2-grams: `aa` to `zz`),
- Numerical n-grams (e.g., for 2-grams: `00` to `99`),
- Alphanumeric mixed n-grams (e.g., for 2-grams: `a0` to `z9`).

Since the datasets used in this research primarily contain first names, last names, and birthdates, these character sets are sufficient to cover the vast majority of n-gram occurrences. Each

possible n-gram is mapped to a specific index in the output tensor based on the dictionary, ensuring a consistent label format across all training samples. For example, if index `1` corresponds to the n-gram "ab", and the Artificial Neural Network predicts a 60% probability at index `1`, this is interpreted as a 60% likelihood that "ab" was present in the original plaintext.

By structuring the data in this way, the Artificial Neural Network is trained to learn a mapping from encoded inputs to their corresponding n-gram distributions, enabling the Dataset Extension Attack to probabilistically reconstruct plaintext Personally Identifiable Information from encoded data.

## 3.6 Step 3: Hyperparameter Optimization

Hyperparameter tuning plays an important role in achieving optimal model performance. Unlike model parameters that are learned during training (e.g., weights and biases), hyperparameters are defined prior to training and control the structure of the model as well as aspects of the learning algorithm. These include architectural choices such as the number of layers as well as training configurations like the learning rate, optimizer, and regularization techniques. Careful selection of these values is important in tasks such as reconstructing plaintext n-grams from encoded representations, where both underfitting and overfitting can lead to substantial performance degradation.

To explore the extensive hyperparameter space efficiently, this work employs Ray Tune, a scalable library for distributed hyperparameter tuning. Specifically, the Optuna search algorithm is used within Ray Tune to guide the optimization process. Optuna leverages a Tree-structured Parzen Estimator, a Bayesian optimization method that prioritizes promising regions of the search space based on previous trial results. This approach improves search efficiency and reduces the number of iterations required to discover high-performing configurations.

The hyperparameter search space in this study is designed to be both comprehensive and computationally feasible. Key hyperparameters that define the neural network architecture include:

- **Number of hidden layers**: varied between 1 and 4, allowing the exploration of both shallow and deep networks.
- **Hidden layer size**: selected from {128, 256, 512, 1024, 2048}, enabling experiments with compact to large-capacity models.
- **Dropout rate**: sampled uniformly between 0.1 and 0.4 to promote generalization and mitigate overfitting.
- **Activation function**: treated as a categorical variable with options including `ReLU`, `Leaky ReLU`, `GELU`, `ELU`, `SELU`, and `Tanh`. All of these functions introduce non-linearity and aim to mitigate vanishing gradients, but differ in smoothness, output range, and handling of negative inputs.

These architectural parameters create a flexible and expressive search space for discovering well-performing network structures tailored to the task of the Dataset Extension Attack.

The optimization strategy is similarly governed by several hyperparameters that influence how the model is trained. The **optimizer** is treated as a categorical hyperparameter, with options including `Adam`, `AdamW`, `RMSprop`, and `SGD`. Each optimizer is paired with a corresponding

learning rate sampled from a log-uniform distribution to accommodate the wide sensitivity of models to this parameter. In the specific case of `SGD`, an additional `momentum` parameter is also tuned to control the influence of past gradients in the current weight update.

In conjunction with the optimizer, the choice of a **learning rate scheduler** further enhances the model's ability to converge effectively. The search space for learning rate scheduling strategies includes:

- `StepLR`: reduces the learning rate at fixed epoch intervals,
- `ExponentialLR`: applies exponential decay over time,
- `ReduceLROnPlateau`: reacts to stagnation in validation loss,
- `CosineAnnealingLR`: follows a cosine decay schedule,
- `CyclicLR`: oscillates between lower and upper bounds in modes such as `triangular`, `triangular2`, and `exp_range`.

An additional option to disable learning rate scheduling is also included to assess whether constant learning rates perform better for certain models.

Furthermore, the **loss function** is a hyperparameter, as it directly influences the optimization objective. Several loss functions suitable for multi-label classification are explored:

- `BCEWithLogitsLoss`: a standard binary cross-entropy loss combined with a sigmoid activation, commonly used for multi-label tasks.
- `MultiLabelSoftMarginLoss`: supports probabilistic multi-label targets and is well-suited for scenarios like the Dataset Extension Attack where multiple n-grams are present simultaneously.
- `SoftMarginLoss`: a generalization of logistic loss for binary classification, which can also be applied to multi-label settings with continuous targets.

This comprehensive optimization configuration enables systematic exploration of training dynamics, ensuring the neural network can effectively learn meaningful mappings for the Dataset Extension Attack across various encoding schemes.

**Additional parameters** are included in the hyperparameter search to fine-tune the model's output behavior and ensure consistent evaluation. A tunable **threshold** parameter, ranging from 0.3 to 0.8, is introduced to convert the model's probabilistic outputs into binary predictions for the presence of specific n-grams. This threshold plays a role in multi-label classification, as it directly affects the balance between precision and recall. Furthermore, the **batch size** used by the data loaders is treated as a tunable parameter, with candidate values of 8, 16, 32, and 64. Varying the batch size allows for a trade-off between computational efficiency and training stability, potentially influencing convergence dynamics and generalization performance.

To ensure a fair and consistent comparison across all hyperparameter configurations, the same training and validation datasets are used in each trial. This controlled setup ensures that variations in performance can be attributed to the model configuration rather than differences in training and validation data.

During tuning, Ray Tune orchestrates multiple parallel trials, each corresponding to a unique combination of hyperparameters sampled from the search space. Optuna's pruning mechanism

is also integrated, allowing unpromising trials to be stopped early based on intermediate results (e.g., validation loss), which improves overall efficiency. Performance is evaluated on the validation set, and the best configuration is selected based on a predefined optimization metric, as discussed in Section **??**.

This automated, systematic tuning process ensures that the neural network architecture is well-adapted to the complexity and characteristics of the input encoding. It enables fair comparison across models and improves both the predictive performance and generalization capability of the reconstruction task.

## 3.7 Step 4: Model Training and Artificial Neural Network Architecture

The architecture follows a feedforward design and consists of three main components: an input layer, a configurable sequence of hidden layers, and a final output layer. The size of the input layer is determined by the dimensionality of the encoded record. For instance, in the case of the Bloom Filter and Tabulation MinHash model, the input layer corresponds to the length of the bitstring, which in turn is defined by Alice's chosen parameters. The Two-Step Hash models define their input layers based on the number of unique integers used across both datasets, the re-identified and non-re-identified individuals.

This modular architecture enables experimentation across different encoding schemes while maintaining a unified framework for training and evaluation.

The hidden layers are structured dynamically, depending on a set of tunable hyperparameters. These include the number of hidden layers and the number of neurons in each layer (hidden layer size). Each hidden layer is followed by a non-linear activation function, enabling the model to capture complex and non-linear relationships in the data. To mitigate overfitting, dropout is applied after each activation layer, randomly deactivating a fraction of neurons during training. This regularization technique improves the model's ability to generalize to unseen inputs and prevents the memorization of training data.

The output layer remains consistent across all encoding schemes and has a dimensionality equal to the size of the predefined n-gram dictionary. Each output neuron represents the model's predicted probability that a specific n-gram appears in the original plaintext record. Given that multiple n-grams may be present in a single encoded record, the task is framed as a multi-label classification problem. Therefore, the output layer uses a sigmoid activation function, allowing the network to assign independent probability estimates to each n-gram.

### 3.7.1 Foundations of Neural Network Success in Dataset Extension Attack

Attempting to reconstruct plaintext information from encoded representations based on hash functions presents a challenge due to the nature of cryptographic hashing. Since hash functions are designed as one-way functions, reversing the transformation to recover the original input is theoretically infeasible. However, while exact reconstruction is not possible, a probabilistic approach can still be employed to infer likely plaintext components based on statistical patterns within the encoded data.

Artificial Neural Networks provide a framework for learning complex mappings between input encodings and output predictions, making them well-suited for this task. The function of the Artificial Neural Network in the context of the Dataset Extension Attack is to predict n-grams by learning from re-identified individuals, those whose plaintext information is known alongside their corresponding encodings. Through this supervised learning process, the model captures

frequency patterns that emerge due to the deterministic nature of the encoding process. In essence, the Artificial Neural Network learns which n-grams are statistically associated with specific positions or patterns in the encoded representations and leverages these associations to estimate their likelihood in unseen encoded inputs.

Although hash functions introduce collisions, where different inputs may produce the same hash output, the Artificial Neural Network can still extract meaningful probabilistic insights by generalizing over these mappings across many samples. This enables the Dataset Extension Attack to output a ranked list of likely n-grams per record, thereby forming the basis for the reconstruction of Personally Identifiable Information from encoded data in a probabilistic, frequency-informed manner.

This is possible due to the fact that for names certain n-grams are more likely to occur than others. For example, n-grams like "an", "el", or "ar" are common in many names, while others like "xz" or "qv" are rare. By learning these statistical patterns, the Artificial Neural Network can prioritize more probable n-grams during reconstruction, improving the overall accuracy of the attack.

### 3.7.2 Training the Model

To effectively train and evaluate the Artificial Neural Network model with the best hyperparameters, the dataset is divided into three distinct subsets: a training set, a validation set, and a test set. The training set consists of 80% of the labeled dataset, while the remaining 20% is designated as the validation set. The test set comprises the non-re-identified individuals, serving as the primary evaluation set for the trained model.

Dataloaders are created for each of these subsets to facilitate efficient mini-batch processing. Different batch sizes are employed depending on the chosen hyperparameter value to optimize computational performance and convergence behavior. The training and validation dataloaders enable efficient iteration over the respective data splits, ensuring that the Artificial Neural Network is exposed to all available samples during training and validation.

The training process consists of multiple epochs, where each epoch involves iterating through the entire training dataset using the data loader. For each mini-batch, the model performs a forward pass, computes the loss, and applies backpropagation to update the network's parameters using the selected optimizer. If a learning rate scheduler is specified, it is applied according to its strategy to dynamically adjust the learning rate throughout training.

After processing all training batches in an epoch, the model's performance is evaluated on the validation set by computing the validation loss and selected performance metrics. This allows the training loop to monitor potential overfitting and adjust training accordingly, for instance by applying early stopping or learning rate decay. Throughout this process, the model learns the optimal weights and biases that minimize the loss on the training set, guiding it toward better generalization and performance.

## 3.8 Step 5: Application to Encoded Data

### 3.8.1 Performance Evaluation

Once the Artificial Neural Network is trained and validated, it can be applied to the non-re-identified individuals, whose encoded representations were not matched during the Graph Matching Attack step. The performance of the Dataset Extension Attack is evaluated on

this test set. Several metrics are computed to assess the effectiveness of the attack, including precision, recall, F1-score, and the Dice similarity coefficient.

**Precision** quantifies the proportion of correctly predicted n-grams among all n-grams predicted by the model. It reflects the model's ability to avoid false positives during the reconstruction of plaintext features from encoded representations. A high precision value indicates that most of the predicted n-grams are indeed part of the original Personally Identifiable Information, suggesting a low rate of over-generation. This is particularly relevant in the context of Dataset Extension Attacks, where the specificity of reconstructed values is to minimize erroneous inferences. Formally, for a given record, let $T$ be the set of true n-grams and $P$ the set of predicted n-grams. The precision is defined as:

$$\text{Precision} = \frac{|T \cap P|}{|P|} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

where TP denotes the number of true positives and FP the number of false positives. If $|P| = 0$, precision is defined as 0 to avoid division by zero.

**Recall** measures the proportion of true n-grams that were successfully predicted by the model. It captures the completeness of the reconstruction, indicating how much of the original information was extracted from the encoded data. A high recall implies that the attack is capable of recovering a fraction of the original content, even if some incorrect n-grams are also included. The recall is defined as:

$$\text{Recall} = \frac{|T \cap P|}{|T|} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

where FN represents the number of false negatives (i.e., true n-grams missed by the model). If $|T| = 0$, recall is defined as 0.

**F1-score** is the harmonic mean of precision and recall, providing a single score that balances both correctness and completeness. It is informative in scenarios where the number of predicted and ground-truth n-grams may differ substantially, preventing either precision or recall from dominating the performance evaluation. The F1-score is defined as:

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

If both precision and recall are zero, the F1-score is defined as zero.

The **Dice similarity coefficient**, which is mathematically equivalent to the F1-score for binary sets, is computed as twice the size of the intersection of the predicted and actual n-gram sets divided by the total number of elements in both sets. In the Dataset Extension Attack setting, the Dice coefficient serves as an interpretable and robust measure of set overlap. It is well-suited for evaluating partial reconstructions, where perfect recovery may be infeasible, but alignment with the ground truth still reflects successful inference.

Each of these metrics is computed on a *per-record* basis, comparing the predicted and true n-gram sets for every individual sample. To evaluate the model's performance on the entire dataset, the scores are then averaged across all records. This approach ensures a granular and interpretable assessment of the reconstruction quality, accounting for varying degrees of difficulty across different entities.

### 3.8.2 Choosing the Right Metric for Hyperparameter Optimization

The performance of the neural network employed in the Dataset Extension Attack is evaluated using several metrics: precision, recall, F1-score, and the Dice similarity coefficient. While each metric provides valuable insight into the quality of plaintext reconstruction, only one can be selected as the optimization target during hyperparameter tuning. From the attacker's perspective, this choice is strategic, as it directly influences the re-identification behavior and the nature of the reconstructed information.

**Optimizing for Precision: Conservative but High-Confidence Inference** Optimizing for precision encourages the model to prioritize correctness over completeness. In this configuration, the model is incentivized to predict fewer n-grams but with high certainty that each predicted token is truly part of the original plaintext. This reduces the risk of false positives, which is important when re-identified individuals inform downstream actions, such as attempting identity theft or inferring sensitive attributes. A precision-oriented attacker thus obtains fewer, but more reliable, re-identifications, minimizing noise in the reconstructed dataset. However, this strategy may overlook harder-to-recover but valid features.

**Optimizing for Recall: Aggressive and Broad Coverage** In contrast, recall optimization emphasizes completeness, seeking to recover as many original n-grams as possible, even at the cost of accuracy. This strategy is advantageous when the attacker aims to maximize information gain, such as in exploratory analysis or probabilistic linkage. High recall ensures broader coverage of the encoded dataset and may benefit post-processing or correction steps filtering out wrongly predicted n-grams. However, it also increases the risk of false positives, potentially degrading the overall quality of re-identifications.

**Optimizing for F1-Score or Dice: Balanced Inference** The F1-score and the Dice similarity coefficient (which are mathematically equivalent for binary sets) provide a harmonic balance between precision and recall. Optimizing for these metrics helps prevent predictions that are either too conservative or too permissive. Since the attacker typically cannot externally validate predictions, balancing precision and recall mitigates the risk of errors. Additionally, the Dice coefficient provides an intuitive and interpretable measure of similarity,

**Strategic Consideration** Ultimately, the attacker's choice of optimization metric should reflect the intended downstream use of the reconstructed data. If re-identified records are used to trigger sensitive actions, such as identity theft or inferring information about an person, then precision is the preferred metric, as it minimizes the risk of false positives. In contrast, when the goal is to maximize overall data leakage or to support probabilistic analyses, recall becomes more advantageous, as it emphasizes completeness over certainty. For scenarios that require a balanced evaluation of both precision and recall, particularly when no strong bias toward one type of error exists, metrics such as the F1-score or Dice coefficient provide a robust and interpretable compromise.

In this study, the Dice similarity coefficient was selected as the primary optimization objective during hyperparameter tuning, due to its balanced nature and its interpretability in the context of set overlap. This reflects the attacker's goal of performing broadly effective and consistent reconstructions, maximizing utility while minimizing both false positives and false negatives.

## 3.9 Step 6: Refinement and Reconstruction

The subsequent step focuses on reconstructing interpretable plaintext attributes, such as first name, surname, and date of birth, from the sets of n-grams predicted by the Artificial Neural Network following a threshold-based filtering process. This reconstruction is essential for empirically demonstrating the extent to which encoded identifiers can be reversed into human-readable information, thereby highlighting the problem of privacy vulnerabilities in Privacy-Preserving Record Linkage systems.

To tackle this task, three distinct reconstruction strategies are explored, each offering different trade-offs in terms of computational complexity, accuracy, interpretability, and the level of certainty regarding the correctness of the reconstructed identifiers. These strategies are designed to simulate varying levels of attacker sophistication, from basic structural heuristics to dictionary guided and Machine Learning assisted inference, thereby providing a comprehensive perspective on the practical threat posed by Dataset Extension Attack.

The first method, a graph-based approach, serves as a structural baseline and operates in a fully deterministic manner. It constructs a directed graph in which each node represents a character, and each edge corresponds to a predicted n-gram that links the starting character to the ending character. The reconstruction task is framed as finding the longest paths through this graph, under the assumption that sequences having the greatest number of predicted n-grams are most likely to represent complete and meaningful strings. These paths are then interpreted as candidate identifiers.

This approach is computationally efficient and independent of external resources, but it does not resolve ambiguities when multiple equally long or plausible paths exist. Moreover, it lacks semantic validation mechanisms to verify whether the reconstructed substrings resemble real-world values, which limits its practical effectiveness in realistic attack scenarios. On the other hand, it offers the highest certainty regarding reconstruction correctness among the evaluated methods, as it relies solely on the predicted n-grams without introducing any additional inference or external assumptions. This makes it a transparent and interpretable technique, particularly suitable for isolating the contribution of the n-gram prediction model itself.

The second approach incorporates prior knowledge through a dictionary-based fuzzy matching technique. A curated reference list of known names, comprising frequently occurring first names and surnames, is preprocessed into corresponding sets of n-grams. These name lists are sourced from the U.S. Census and the Social Security Administration, providing realistic and demographically representative coverage of common naming patterns. For each predicted set of n-grams, candidate names from the dictionary are evaluated using similarity metrics such as the Dice or Jaccard coefficient. The top-scoring candidates are selected as reconstruction hypotheses, prioritizing entries with the greatest n-gram overlap.

This method offers increased interpretability by anchoring the reconstruction process in semantically valid, real-world values and provides a degree of robustness against structural noise or incomplete n-gram sets. However, its effectiveness is inherently limited by the quality and coverage of the dictionary. It may fail to reconstruct rare or out-of-vocabulary names and implicitly assumes that personal names adhere to those found in the predefined list.

The third and most flexible strategy employs a generative language model to reconstruct attributes directly from the set of predicted n-grams. In this approach, the unordered collection of n-grams is embedded into a natural language prompt, allowing the model to infer likely combinations of names and birthdates. Leveraging its ability to synthesize coherent text and recognize semantic relationships, the language model can resolve ambiguities, infer structural

patterns, and even compensate for missing or noisy n-grams.

While this method offers the greatest expressive power and adaptability, it also introduces notable limitations in terms of reproducibility, transparency, and correctness. The model may produce outputs that are fluent but factually incorrect (i.e., hallucinations), and there is no guarantee that all input n-grams are represented in the reconstructed response. As such, this approach is best employed as a heuristic aid to guide human interpretation, rather than as a deterministic component of the reconstruction pipeline.

Together, these three reconstruction strategies offer complementary perspectives: the first illustrates the structural feasibility of assembling plausible tokens from n-gram fragments, the second emphasizes semantic validation through prior knowledge, and the third showcases the generative capabilities of modern language models. A comparative evaluation of their outputs enables a nuanced assessment of the threats posed by Dataset Extension Attacks, illustrating how attackers with varying levels of knowledge, computational resources, and methodological assumptions could effectively compromise encoded personal data.

### 3.9.1 Directed Graph Based Reconstruction

The first reconstruction strategy is based on modeling the set of predicted n-grams as a directed graph to recover plaintext attributes. In this representation, each character becomes a node, and each n-gram is modeled as a directed edge from its first character to its last. The resulting graph is typically simple, though it can be either acyclic or cyclic depending on the structure and overlap of the predicted n-grams. Reconstruction is then formulated as a longest path problem within this graph. The central assumption is that the longest possible path, in terms of included n-grams, is most likely to represent the most complete and coherent reconstruction of the original plaintext string.

In the case of a Directed Acyclic Graph, the longest path can be computed efficiently using well established graph traversal algorithms. Specifically, this implementation leverages NetworkX's built-in `dag_longest_path` function, which operates in linear time with respect to the number of nodes and edges ($\mathcal{O}(V + E)$), making it suitable for small to moderately sized graphs typical in this setting.

When the graph contains cycles, however, the problem becomes more complex, as naive traversal may result in infinite loops. To handle this, a custom recursive Depth First Search Search strategy is employed. The algorithm explores all reachable paths from each starting point while maintaining a set of visited edges to avoid revisiting the same n-gram multiple times within a single path. At each step, the current reconstruction string is extended with the next character, and the longest sequence encountered during traversal is stored.

An illustrative example is shown in Figure **??**, based on the set of 2-grams `["jo", "oh", "hn", "do", "oe", "be", "ha", "af", "st"]`, which forms a directed graph with eleven nodes. Since the graph is cyclic, the algorithm performs a recursive Depth First Search, tracking visited edges to prevent infinite loops. It initiates the search from each node, extending candidate paths by appending the next character at each step. The longest path found in this process corresponds to the string "johndoe", as all alternative paths result in shorter sequences.

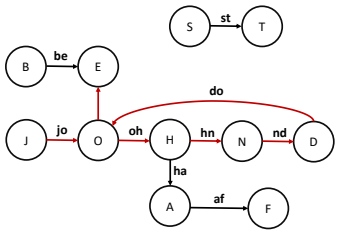


**Figure 3.2:** Example reconstruction using the Directed Graph based approach.

This method is fully deterministic and does not rely on any external data sources, making it flexible and interpretable. However, it still faces challenges in resolving ambiguities when multiple equally long paths exist and lacks any semantic validation to ensure that the output resembles real-world names or values. Nonetheless, it provides a strong structural baseline that emphasizes maximal use of the predicted n-gram set.

From a computational perspective, this approach is efficient for small graphs, as typically encountered in n-gram-based reconstructions of individual fields such as names. For Directed Acyclic Graphs, the runtime remains linear, and even in the presence of cycles, the Depth First Search based implementation remains tractable due to the limited number of characters and n-grams per input record. In practice, most graphs contain fewer than 30 edges, and recursion depth is shallow. The approach scales well to batch processing across many records and offers a favorable trade-off between accuracy and runtime, making it a practical first stage reconstruction strategy.

### 3.9.2 Dictionary Based Reconstruction

The dictionary based reconstruction strategy leverages curated datasets of common attributes like first names, surnames, and birthdates to infer plaintext attributes from predicted n-grams. For each predicted record, the method computes a similarity score between the predicted set of n-grams and the n-gram representations of each dictionary entry. This is performed separately for each attribute, e.g. first name, surname, and birthdate, by comparing the predicted n-grams to those extracted from the corresponding dictionaries. The top scoring candidates are selected as reconstruction hypotheses, prioritizing entries with the greatest n-gram overlap as measured by a similarity metric, such as the Dice coefficient.

This approach increases interpretability by grounding predictions in semantically valid and human readable values. It is robust against noise or partial n-gram sets, as it can still return plausible results even when only a subset of the correct n-grams is present. Moreover, by using separate dictionaries for each attribute, the method is capable of multi-field reconstruction, making it more versatile and scalable than methods that only target a single attribute.

Reconstruction is conducted sequentially across multiple attributes. First, the predicted n-grams are matched e.g. against a list of known first names. Once the most likely match is found, its n-grams are removed from the candidate set to avoid double counting when reconstructing the next attribute. This is particularly useful in cases where certain attributes like names can

appear both as first and last names (e.g., "James"). A similar strategy is then applied to the remaining attributes.

The quality of this reconstruction method is highly dependent on the coverage and granularity of the dictionaries used. Larger and more representative dictionaries improve the likelihood of accurate matches, while rare or culturally diverse names may remain underrepresented. Additionally, the similarity scoring mechanism plays a key role. The Dice coefficient is used due to its effectiveness in measuring set overlap between short strings, such as n-gram sets.

The runtime complexity of the dictionary-based reconstruction strategy is primarily determined by the number of predicted entries and the size of the reference dictionaries.

For each of the $n$ predicted entries, the algorithm performs similarity comparisons against all candidates in three separate dictionaries: given names, surnames, and birthdates. Let $D_{\text{given}}$, $D_{\text{surname}}$, and $D_{\text{birthday}}$ denote the number of entries in each respective dictionary. The total number of comparisons per entry is then

$$D = D_{\text{given}} + D_{\text{surname}} + D_{\text{birthday}}.$$

The similarity metric used for comparison is the Dice coefficient over n-grams, which can be computed in linear time with respect to the number of n-grams in the compared tokens. Assuming an average of $g$ n-grams per comparison (i.e., $|A| + |B| \approx g$), the runtime complexity of computing a single Dice similarity is $\mathcal{O}(g) = \mathcal{O}(|A| + |B|)$ where $|A|$ and $|B|$ are the number of n-grams in the two sets being compared.

Therefore, the overall runtime complexity of the dictionary-based reconstruction attack is

$$\mathcal{O}(n \cdot D \cdot g)$$

.

In practice, the values of $g$ are relatively small, as the average number of 2-grams per name is limited (typically between 4 and 10). Therefore, the reconstruction remains efficient even for large $n$. Furthermore, this approach is trivially parallelizable across entries, making it well suited for batch processing in realistic attack scenarios.

Overall, this strategy provides a realistic and reproducible method for attackers to reconstruct personal information, particularly in scenarios where the attacker has access to auxiliary data such as population wide name lists or public birthdate datasets. By exploiting structural patterns and common token distributions, this method demonstrates how dictionary guided attacks can enhance the re-identification capabilities.

### 3.9.3 Generative Language Model Based Reconstruction

The final and most flexible reconstruction strategy explored in this thesis involves leveraging Large Language Model to infer original identifiers from predicted n-grams. Unlike previous approaches, which operate within constrained matching logic, Large Language Models can reason over partial, noisy, or ambiguous input and generate semantically coherent completions based on prior knowledge. In this context, the model is prompted with a batch of predicted n-gram sets and asked to reconstruct corresponding attribute values such as given names, surnames, or birthdates.

This approach proves particularly robust in cases where the n-gram predictions are incomplete or include noise. Owing to their generative and contextual capabilities, Large Language

Models can infer plausible attribute values even when important information is missing. Additionally, the model can implicitly correct for common errors or insert culturally plausible completions, making it attractive for reconstructing personal identifiers under uncertainty. This flexibility, however, comes at the cost of reproducibility and transparency. Since Large Language Models are non-deterministic, their responses may vary across repeated executions, even with identical input. Furthermore, hallucinations, plausible but incorrect outputs, can occur when the model overgeneralizes or encounters ambiguous prompts.

Another practical limitation concerns model availability and cost. If the attacker does not have access to a self-hosted Large Language Model, they must rely on external APIs (e.g., OpenAI or similar providers), which introduces cost and latency. Moreover, results can vary substantially depending on the specific model used (e.g., GPT-3.5, GPT-4, open-source variants) and on the design of the prompt. To address the efficiency of batch processing, this thesis adopts a strategy in which 15 n-gram sets are reconstructed in parallel per prompt call, a trade-off found to balance cost, latency, and response quality effectively.

The prompt used for this reconstruction task is designed to guide the model toward structured attribute extraction while accommodating the free-form nature of language generation. The prompt template can be found below and should be considered an integral part of the reconstruction method:

> **Simplified Prompt Template:**
>
> You are an attacker attempting to reconstruct the **given name**, **surname**, and **date of birth** from a set of 2-grams for each individual.
>
> Each entry contains:
>
> - `<GIVEN_NAME>`
> - `<SURNAME>`
> - `<BIRTHDATE>` (format: `M/D/YYYY`, no leading zeros)
>
> Return a valid JSON array with the reconstructed information for each individual, including their unique identifier, given name, surname, and date of birth.
>
> The input format consists of a mapping from unique identifiers to lists of 2-grams extracted from the encoded records.

After submitting the prompt, the model typically responds within several seconds, depending on the provider and load conditions. The output is then post-processed to extract the structured attribute values. While this approach is not used in the final evaluation due to its lack of reproducibility and its dependency on external infrastructure, it represents a direction for future attack strategies, especially in human-in-the-loop or investigative contexts where creative reconstruction is desirable.

# 4 Results

## 4.1 Experiments

To evaluate the effectiveness of the previously defined Dataset Extension Attack, a series of experiments are conducted using multiple datasets. These experiments aim to assess the attack's performance across different encoding schemes and datasets using different execution settings to analyze its ability to reconstruct plaintext information from encoded identifiers.

The primary dataset used is the `fakename` dataset, which is synthetically generated using the American name set provided by the Fake Name Generator. This dataset was previously employed in related work by Schaefer et al. [SAH24], making it a suitable benchmark for comparative evaluation. It includes realistic combinations of personal identifiers and is well-suited for testing the scalability and reliability of both the Graph Matching Attack and Dataset Extension Attack pipelines.

The `fakename` datasets consist of synthetically generated entries, each containing a given name, surname, and date of birth. These datasets aim to resemble realistic combinations of personal identifiers while ensuring privacy and reproducibility. For evaluation purposes, multiple dataset instances of varying sizes are used: 1,000, 2,000, 5,000, 10,000 and 20,000 entries.

The primary advantage of using this dataset family lies in its scalability. By maintaining a consistent schema while varying the number of records, the impact of dataset size on the performance and success of the Dataset Extension Attack can be systematically analyzed. This enables controlled experiments that highlight how the quantity of available data influences re-identification, training quality, and generalization performance of the attack models.

An additional dataset used in this study is the `euro_person` dataset provided as part of the simulated data for the ESSnet DI on-the-job training course on record linkage. The dataset was created by Paula McLeod, Dick Heasman, and Ian Forbes from the UK Office for National Statistics and contains realistic, fictionalized personal information intended for the training and evaluation of record linkage techniques. The `euro_person` dataset includes forename (`PERNAME1`), surname (`PERNAME2`), and full date of birth composed of day, month, and year, which were concatenated into a single `DOB_FULL` attribute for the purposes of this work. The dataset consisting of 26.625 records also serves as a ground-truth reference for other simulated sources such as Census, CIS, and PRD, it is well-suited for evaluating the precision and completeness of plaintext reconstruction and re-identification in the context of the Dataset Extension Attack.

In addition to the synthetic and benchmark datasets, this thesis also incorporates a curated version of the Titanic passenger manifest, referred to as `titanic_full`. This dataset consists of 891 unique records and includes the fields `firstname` and `surname`. While not originally intended for record linkage evaluation, the dataset offers a semi-realistic collection of personal identifiers derived from historical records. It provides a useful case for examining the impact of natural name diversity, varying name lengths, and non-standard naming formats (e.g., inclusion of titles or parenthetical information) on the performance of both Graph Matching Attack and

Dataset Extension Attack. Due to the historical and English centric nature of the data, it shares some limitations with other Western-focused datasets used in this work but nonetheless adds valuable variety in terms of name structure and frequency.

The experiments are conducted across a range of settings and scenarios to comprehensively evaluate the effectiveness of the Dataset Extension Attack. Each encoding scheme, namely Bloom Filter, Two-Step Hash, and Tabulation MinHash, is tested individually across all datasets as described earlier. This allows for a direct comparison of reconstruction performance under different privacy-preserving encoding mechanisms.

To ensure consistent and comparable evaluation across all encoding schemes, standardized configuration parameters were employed for both Alice's and Eve's encoding processes. All three encoding schemes used 2-grams with Dice similarity as the distance metric, providing a common baseline for structural comparison. For Bloom Filter encoding, both parties used a filter length of 1024 bits with 10 hash functions and no diffusion layer, ensuring maximum compatibility while maintaining reasonable privacy guarantees. Tabulation MinHash configurations employed 1024 hash functions with 64-bit precision, 8 sub-keys, and 1-bit hash output, balancing computational efficiency with encoding quality. Two-Step Hash settings utilized 10 hash functions with 1000 columns and Pseudo-Random Number Generator-based randomization, providing sufficient dimensionality for effective similarity preservation. These standardized parameters ensure that performance differences observed in the experiments can be attributed to the fundamental characteristics of each encoding scheme rather than configuration-specific optimizations.

To additionally analyze how the quantity of the training data affects the Dataset Extension Attack, the preceding Graph Matching Attack step is executed with varying levels of overlap between Alice's and Eve's datasets. For each dataset and encoding scheme, the Graph Matching Attack is run multiple times with overlap ratios ranging from 20% to 80%, in increments of 20%. This simulates different real world scenarios where the attacker has access to varying amounts of auxiliary information. The resulting re-identifications from the Graph Matching Attack then serve as the labeled training data for the Dataset Extension Attack, thus allowing for a detailed evaluation of how overlap levels influence overall reconstruction success.

In addition to varying dataset sizes and overlap levels, different attacker scenarios are considered by employing different drop from strategies to evaluate the robustness of the Dataset Extension Attack under more and less realistic assumptions. The first scenario, Eve's auxiliary dataset $D_p$ is a strict subset of Alice's dataset $D_e$, i.e., $D_p \subseteq D_e$. In this case, the overlap $o$ is defined as the ratio $o = \frac{|D_p|}{|D_e|}$. The elements in $D_p$ are generated by randomly sampling $|D_p| = \lfloor o \cdot |D_e| \rfloor$ records from $D_e$ without replacement. While this setup simplifies evaluation and isolates the impact of training data availability, it is also highly idealized and does not reflect the complexity of real world linkage scenarios.

To address this, a second, more realistic setting is also considered, where both $D_p$ and $D_e$ contain disjoint as well as overlapping individuals. That is, $D_e \nsubseteq D_p$, but $D_e \cap D_p \neq \emptyset$. In this scenario, the auxiliary and target datasets each include individuals not present in the other, simulating cases where Eve has partial but non exclusive knowledge of the data. This setup introduces additional challenges for both the Graph Matching Attack and Dataset Extension Attack, as structural mismatches and auxiliary noise may degrade re-identification and reconstruction accuracy.

This setup mirrors the experimental methodology employed by [SAH24], ensuring consistency and comparability with prior work on the Graph Matching Attack. By varying the

overlap rate and dataset composition in this way, a diverse range of re-identification scenarios is created, which directly impacts the amount and quality of training data available for the Dataset Extension Attack. This, in turn, enables a systematic evaluation of the Dataset Extension Attack's ability to generalize from partially re-identified data.

As the Graph Matching Attack identifies different subsets of individuals under varying overlap conditions, the resulting re-identification sets are used to train the neural network, while the remaining non matched records serve as the test set. Thus, each experiment yields a distinct train-test split, providing a rich basis for assessing the reconstruction capabilities of the Dataset Extension Attack.

For the Dataset Extension Attack specific configuration, several fixed settings were employed to ensure comparability across all experimental conditions. First, the dataset of re-identified individuals, used as labeled training data, was split into training and validation sets using a fixed 80/20 ratio. This choice reflects common Machine Learning practice and provides a balanced compromise between model learning and validation reliability.

One of the most critical components of the Dataset Extension Attack pipeline is the hyperparameter optimization step, which is responsible for identifying the most effective neural network architecture. For this purpose, a total of 125 trials were conducted for each experimental setting. This number was chosen to provide sufficient coverage of the hyperparameter space while maintaining computational feasibility.

Each trial, as well as the final training run for the best performing model, was limited to a maximum of 20 training epochs. While this represents a relatively high upper bound, overfitting is mitigated through the use of early stopping. Specifically, training was halted if the validation loss did not improve for five consecutive epochs (patience = 5), with a minimum delta of $1 \cdot 10^{-4}$ required to qualify as an improvement. This strategy ensures both efficient training and effective model selection, especially when performance plateaus early.

The search space for the hyperparameter optimization follows the configuration described in Section **??**. Throughout the entire Dataset Extension Attack pipeline, the **Dice coefficient** is used as the objective metric for optimization. This choice is motivated by its robustness and balanced nature, as it integrates both precision and recall and has consistently yielded the most promising results during preliminary manual testing.

For efficient optimization, the hyperparameter search is executed using all available CPU cores. Additionally and optionally an NVIDIA GPU can be used to accelerate the training of the neural networks during hyperparameter optimization. This allows for near maximal parallelism during hyperparameter tuning, reducing the total runtime of the experiments.

In the final re-identification phase, two reconstruction strategies are evaluated to enable comparative analysis: (1) the greedy, graph-based reconstruction method described in Section **??**, and (2) the dictionary-based fuzzy matching approach described in Section **??**. Both methods are deterministic and computationally efficient, making them suitable for large scale experimental evaluation.

The language model based reconstruction method is excluded from the evaluation. Despite showing potential in early qualitative testing, its dependence on external models, token-based pricing, and limited reproducibility make it unsuitable for scalable and reproducible experimentation within the current research setting.

All experiments were conducted on a virtual machine running Ubuntu 24.04, equipped with 20 cores of a virtualized AMD processor (QEMU Virtual CPU). The system was provisioned with 176 GB of RAM and featured an NVIDIA GeForce RTX 3090 Ti GPU with 24 GB of dedicated VRAM.

This high-performance computing setup enabled efficient parallel execution of the hyperparameter optimization trials and accelerated training of the neural networks via GPU. The extensive memory capacity was particularly beneficial during dataset preprocessing and batch-wise loading of large datasets, ensuring that all encoding schemes and reconstruction strategies could be evaluated without resource bottlenecks.

## 4.2 Evaluation Metrics

The performance of the Dataset Extension Attack is assessed using several metrics that are systematically recorded throughout the experimental runs.

Although the Dataset Extension Attack constitutes an offline attack, meaning the attacker can operate without time constraints once both encoded datasets are available, the runtime of the attack remains a valuable indicator of its practical feasibility. Therefore, the total runtime as well as the runtime of each individual stage within the Dataset Extension Attack pipeline (e.g., data preprocessing, model training, hyperparameter optimization, inference, and reconstruction) is measured and documented.

A central metric for assessing the effectiveness of the attack is the *re-identification rate*. This metric is defined as the number of individuals that are successfully and correctly re-identified by the Dataset Extension Attack, divided by the total number of individuals who were not matched during the initial Graph Matching Attack.

A successful re-identification in this context means that the Dataset Extension Attack is able to fully reconstruct the original plaintext attributes of a record such that it exactly matches a record in Alice's encoded dataset. Thus, the re-identification rate reflects the proportion of previously unmapped individuals that the attacker can recover using the inference based approach.

Another important aspect of evaluating the Dataset Extension Attack is its performance in predicting the correct n-grams, which forms the basis for reconstructing plaintext attributes. To assess the quality of these predictions, several standard classification metrics are considered, namely, precision, recall, and the F1-score.

In the context of this work, the F1-score is of particular interest, as it has shown to be the most reliable metric for quantifying n-gram-level performance. Moreover, it is mathematically equivalent to the Dice similarity coefficient for binary sets, making it not only interpretable but also consistent with the optimization objective used during the hyperparameter tuning stage (cf. Section **??**). This alignment ensures that the evaluation metric reflects the actual optimization goal of the Dataset Extension Attack.

To enable a meaningful comparison, the performance of different Dataset Extension Attack configurations is evaluated not only against each other but also against a baseline strategy. This baseline serves as a lower bound for the expected prediction quality and helps contextualize the improvements achieved through the proposed Dataset Extension Attack attack pipeline.

The baseline approach simulates an attacker who, for each non-re-identified individual, simply predicts the $k$ most frequent n-grams across the entire dataset. The value $k$ is equal to the average length of an entry minus one, as the n-grams are overlapping. This assumes that the attacker has full access to the distribution of n-grams in the encoded dataset, a reasonable assumption in a research setting, where the dataset is known, but less realistic in real-world attacks. Still, it provides a practical lower-bound for evaluation.

An analysis of the `fakename` datasets revealed that the average total length of a full entry,

comprising first name, surname, and date of birth, is approximately 21 characters. Given that 2-grams are used for encoding, this corresponds to roughly 20 overlapping 2-grams per entry. Consequently, the baseline is defined by selecting the top $k = 20$ most frequent 2-grams across the entire dataset and predicting this fixed set for every test record. This naive method disregards any record specific characteristics and instead reflects a dataset wide frequency-based guess, serving as a simple yet informative lower bound for comparison.

The result is a dataset specific set of baseline metrics. These are mainly precision, recall, F1-score, and Dice similarity, against which the Dataset Extension Attack model's predictions can be compared. These baseline metrics vary with dataset size, since the n-gram frequency distribution shifts with the number of records.

As illustrated in Figure **??**, the guessing-based baseline for the `fakename` dataset yields relatively stable performance across increasing dataset sizes. While precision values remain low, around 0.215, recall consistently exceeds 0.245. This results in F1 scores clustering near 0.230, with similarly low Dice similarity scores. Notably, the limited variance across dataset sizes suggests that the baseline's effectiveness is only marginally impacted by the scale of the dataset, despite shifts in n-gram frequency distributions. These findings reinforce the role of the baseline as a simple, size-agnostic lower bound against which more sophisticated, learning-based Dataset Extension Attack models can be benchmarked.

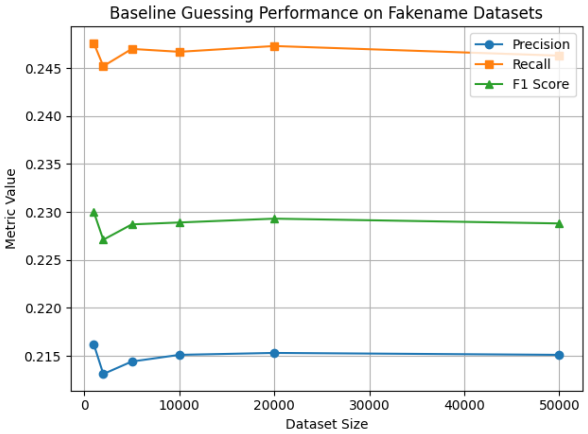


**Figure 4.1:** Evaluation of the baseline performance on the `fakename` dataset: For each dataset size, the prediction quality of the 20 most frequent 2-grams is shown in terms of **precision**, **recall**, and **F1-score**. The average entry length is 21 characters.

For the `euro_person` dataset, the baseline strategy for evaluating the effectiveness of the Dataset Extension Attack follows the same procedure as for the `fakename` dataset. Based on the analysis of this dataset, the average length of a full entry, comprising forename, surname, and full date of birth, is approximately 20 characters. Assuming the use of 2-grams with overlapping character windows, this corresponds to around 19 distinct 2-grams per entry. As a baseline, the top 19 most frequent 2-grams are selected and uniformly predicted for each record, independent of the individual characteristics of the entries.

The resulting performance metrics for this baseline prediction indicate a precision of 0.2197, a recall of 0.2446, and an F1-score of 0.2306. These values provide an reference point for evaluating the added value of the Dataset Extension Attack pipeline, as they quantify the effectiveness of a purely frequency driven reconstruction approach.

For the `titanic_full` dataset, the average length of a full entry (consisting of given name and surname) is approximately 26 characters, indicating relatively high complexity due to longer names. Therefore the top 25 most frequent 2-grams are selected as the baseline for reconstruction. The baseline reconstruction approach achieved a precision of 0.2468, a recall of 0.3770, and an F1 score of 0.2896. These modest performance values reflect the structural challenges posed by the dataset, including the presence of honorifics, compound names, and non-standard formatting. The results suggest that even a naïve guessing strategy can establish a meaningful lower bound for evaluating the performance improvements of learning-based approaches such as the Dataset Extension Attack.

Notably, for the `titanic_full` dataset, the overlap ratio used in the experiments was adjusted to the set {0.2, 0.4, 0.6, 0.7, 0.8, 0.9}. This adaptation was necessary because the dataset is relatively small, and the Graph Matching Attack fails to identify any individuals at lower overlap values. Meaningful results only begin to emerge at an overlap of 0.7 or higher.

Notably, the similarity of the baseline metrics to those obtained on the `fakename` and `euro_person` dataset highlights the generality of the method across synthetic and semi-realistic datasets, further justifying its inclusion as a comparative benchmark.

## 4.3 Analysis

In this section the results of the Dataset Extension Attack experiments are presented and analyzed. The analysis is structured into several subsections, each focusing on a specific aspect of the Dataset Extension Attack performance. As described in Section **??**, the experiments were conducted across multiple datasets, encoding schemes, overlaps and drop from strategies. Therefore, several of these configuration settings will be fixed to analyze the impact of the remaining parameters on the Dataset Extension Attack performance.

One important aspect especially for smaller datasets and overlap sizes is, that it could occur that the Graph Matching Attack does not identify any individuals, i.e., the re-identification set is empty. In this case, the Dataset Extension Attack is not able to reconstruct any plaintext information. This is the case because if there are no re-identified individuals in place, there is no training data for the Dataset Extension Attack available. Therefore the results of these experiments are not reported and the corresponding data points are excluded from the analysis and plots.

The Graph Matching Attack could also fail during the attack by not being able to converge to a stable solution. In this case, the Dataset Extension Attack is not able to reconstruct any plaintext information as well. Therefore the results of these experiments are not reported and the corresponding data points are excluded from the analysis and plots.

In total, 180 experiments were conducted to evaluate the attack's effectiveness under varying conditions, including different overlap levels and drop strategies. On average, the Dataset Extension Attack achieved a re-identification rate of 2.95% and an F1-score of 0.629, with peak values reaching up to 28.75% and 0.992, respectively. Among the encoding schemes, Two-Step Hash exhibited the highest vulnerability with an average re-identification rate of 3.69% and the best structural learnability (F1 = 0.688). In contrast, Tabulation MinHash yielded the lowest average re-identification rate (1.54%) with the lowest structural learnability (F1 = 0.593), indicating greater resilience. Bloom Filter showed a moderate performance with an average re-identification rate of 3.32% and a structural learnability of F1 = 0.606. The following sections analyze these results in detail, focusing on how encoding choices, dataset

characteristics, and attack configurations influence the Dataset Extension Attack's success.

### 4.3.1 Tabulation MinHash

The following subsection focuses on the results obtained using the Tabulation MinHash encoding scheme. In this analysis, each dataset is evaluated under varying settings. To ensure consistency, the dataset and encoding scheme are fixed, while the evaluation focuses on the impact of different overlap sizes and drop-from strategies.

**Titanic Full** On the `titanic_full` dataset, the Dataset Extension Attack achieves stable and moderate performance using Tabulation MinHash encoding. The prediction quality improves consistently with increasing overlap between the auxiliary and target datasets (see Figure **??** in Appendix **??**).

For the `DropFrom = Eve` scenario, the F1-score decreases from 0.74 at overlap 0.7 to 0.17 at overlap 0.8, before recovering to 0.73 at overlap 0.9. This dip at 0.8 is reflected in both recall and precision, suggesting a less effective hyperparameter configuration or reduced data utility at that particular overlap. Precision is generally high across all overlaps, peaking at 0.96.

In the `DropFrom = Both` setting, the Dataset Extension Attack maintains stable growth in both recall and F1-score. Precision remains above 0.78 at all overlaps and reaches 0.96 at overlap 0.8. At 0.9, the F1-score peaks at 0.72, confirming that even with this drop-from strategy, the Dataset Extension Attack can learn meaningful patterns in Tabulation MinHash encoded data.

Despite good predictive performance, no re-identification is achieved under any overlap or matching strategy. Both fuzzy and greedy matching yield a re-identification rate of zero, underscoring the difficulty of exact record recovery on this small and heterogeneous dataset.

In summary, the Dataset Extension Attack demonstrates good structural learning on Tabulation MinHash encoded Titanic data, particularly at higher overlaps, but fails to produce successful record re-identifications in this configuration.

**Fakename 1k** On the `fakename_1k` dataset, the Dataset Extension Attack demonstrates moderate predictive performance when using Tabulation MinHash encoding (see Figure **??** in Appendix **??**). The model's ability to recover n-gram patterns improves with increasing overlap, particularly in the `DropFrom = Eve` setting.

Under `DropFrom = Eve`, F1-scores increase from 0.18 at 0.4 overlap to 0.58 at 0.8 overlap. This trend is mainly driven by improved recall, which rises from 0.10 to 0.43, while precision remains consistently high above 0.92. This indicates that the Dataset Extension Attack makes accurate predictions on a small but increasing portion of the encoded data as more re-identified data becomes available.

The `DropFrom = Both` configuration shows limited performance. Although the F1-score improves from 0.00 to 0.22 between overlaps 0.4 and 0.8, recall stays low (below 0.14), and precision only surpasses 0.85 at the highest overlap. This suggests that the model struggles to generalize well under this drop-from strategy.

Despite some structural learning success, the re-identification rate remains at zero across all overlap values and both drop-from strategies. Neither greedy nor fuzzy matching yields any valid mappings to original records.

Overall, the Dataset Extension Attack on Tabulation MinHash encoded `fakename_1k` data benefits from higher overlap and the `DropFrom = Eve` strategy but remains ineffective for precise record-level re-identification.

**Fakename 2k** On the `fakename_2k` dataset, the Dataset Extension Attack shows a clear improvement in predictive performance with increasing overlap when using Tabulation MinHash encoding (see Figure **??** in Appendix **??**). This trend is especially evident under the `DropFrom = Eve` configuration.

In the `DropFrom = Eve` scenario, the F1-score increases from 0.10 at 0.4 overlap to 0.57 at 0.6, reaching 0.73 at 0.8. This is driven by steady improvements in recall (from 0.06 to 0.62), while precision remains consistently high (above 0.90 at both 0.4 and 0.8). These results suggest that the Dataset Extension Attack learns meaningful structural patterns when sufficient re-identified training data is available.

The `DropFrom = Both` scenario shows much weaker performance. The F1-score only reaches 0.16 at overlap 0.8, with recall staying below 0.11 and precision peaking at just 0.35. This indicates that the `DropFrom = Both` strategy significantly limits the learning signal on smaller datasets, reducing the Dataset Extension Attack's effectiveness.

Re-identification remains unsuccessful across all configurations and overlaps. Neither fuzzy nor greedy matching methods yield any valid re-identifications, which reflects the difficulty of achieving exact plaintext recovery in low-coverage scenarios.

In summary, Tabulation MinHash encoding on the `fakename_2k` dataset enables effective n-gram prediction under the `DropFrom = Eve` strategy, while re-identification remains out of reach under all tested conditions.

**Fakename 5k** For the `fakename_5k` dataset, the Dataset Extension Attack achieves strong performance in reconstructing structural patterns when Tabulation MinHash is used (see Figure **??** in Appendix **??**). Both `DropFrom = Eve` and `DropFrom = Both` settings show substantial improvements in predictive quality with increasing overlap.

In the `DropFrom = Eve` scenario, the F1-score rises from 0.14 at 0.2 overlap to a peak of 0.87 at 0.6. This is supported by a corresponding increase in recall from 0.11 to 0.80, while precision exceeds 0.95 from 0.4 onward. The `DropFrom = Both` configuration follows a similar trend, achieving an F1-score of 0.77 at 0.6 overlap, with precision of 0.98 and recall around 0.65.

Notably, this is the first setting in which the Dataset Extension Attack yields non-zero re-identification rates. In the `DropFrom = Eve` configuration, a re-identification rate of 1.2% is observed at 0.6 overlap, with contributions from both fuzzy and greedy matching. The `DropFrom = Both` setting shows a smaller but non-negligible rate of 0.07% at the same overlap. These results mark the transition from structural to record-level leakage under favorable training conditions.

Overall, this setting demonstrates that the Dataset Extension Attack can become effective at re-identification as dataset size and training overlap increase. Tabulation MinHash encoded Personally Identifiable Information becomes vulnerable to both structural inference and, under certain conditions, full record re-identification.

**Fakename 10k** The Dataset Extension Attack achieves high performance on the `fakename_10k` dataset with Tabulation MinHash encoding (see Figure **??** in Appendix **??**). Both predictive

quality and re-identification effectiveness increase significantly with overlap, particularly under the `DropFrom = Eve` scenario.

In the `DropFrom = Eve` scenario, the F1-score improves from 0.51 at 0.2 overlap to 0.93 at 0.6 and 0.92 at 0.8. Precision remains consistently above 0.96, and recall exceeds 0.9 at the two highest overlaps. The model clearly benefits from access to a larger fraction of re-identified training data, leading to near-perfect structural reconstruction.

The `DropFrom = Both` configuration follows a similar trend but with slightly lower performance at small overlaps. At 0.8 overlap, it reaches an F1-score of 0.91, with precision of 0.98 and recall of 0.86. Compared to smaller datasets, performance differences between the two dropout strategies become marginal at high overlaps.

For the first time, substantial re-identification rates are observed. Under `DropFrom = Eve`, the combined re-identification rate reaches 5.45% at 0.6 overlap and remains above 5% at 0.8. The `DropFrom = Both` configuration achieves a lower peak of 2.97% at 0.8. Both fuzzy and greedy methods contribute to these results, with fuzzy matching consistently recovering more records.

These findings are mirrored in the aggregate relationship between F1-score and re-identification rate, which shows a clear positive correlation. This confirms that structural accuracy translates to successful record-level attacks in larger datasets.

Overall, the Dataset Extension Attack is highly effective on Tabulation MinHash encoded data when training coverage is sufficient and dataset size enables generalization. The combination of high recall and precision leads to significant re-identification risk at overlap levels of 0.6 and above.

**Fakename 20k** The Dataset Extension Attack reaches its highest level of effectiveness on the `fakename_20k` dataset when using Tabulation MinHash encoding (see Figure **??** in Appendix **??**). Both structural reconstruction and re-identification performance scale with dataset size and overlap, confirming the vulnerability of Tabulation MinHash encoded data under Dataset Extension Attacks.

In the `DropFrom = Eve` configuration, F1-scores increase from 0.51 at 0.2 overlap to 0.96 at 0.8. Precision and recall both exceed 0.95 at the highest overlap, indicating that the model is able to reconstruct the n-gram distribution with high fidelity. The `DropFrom = Both` setting follows a similar trajectory, with the F1-score reaching 0.95 at 0.8 and precision nearly identical to the Eve scenario.

The re-identification rates increase sharply with overlap. Under `DropFrom = Eve`, the combined re-identification rate reaches 13.47% at 0.8, with greedy and fuzzy matching each contributing a substantial portion. Even in the more conservative `DropFrom = Both` setting, the rate surpasses 10% at the highest overlap, confirming that re-identification is possible even without access to a clean subset.

The correlation between F1-score and re-identification rate is stronger than in previous datasets, with a visibly steep upward trend. This suggests that small improvements in structural accuracy have increasingly large effects on re-identification success in large-scale datasets.

In summary, the `fakename_20k` experiments demonstrate the full potential of the Dataset Extension Attack. With high overlap and sufficient data, Tabulation MinHash encoding fails to protect against meaningful reconstruction and record-level compromise.

**Europerson** The *euro person* dataset further confirms the vulnerability of Tabulation MinHash encodings under Dataset Extension Attacks (see Figure ?? in Appendix ??). Despite variations in overlap, the Dataset Extension Attack consistently achieves high predictive performance and exhibits meaningful re-identification capability.

Across all overlap values, the model maintains high precision (above 0.92), with only moderate variability in recall. In the `DropFrom = Both` setting, F1-scores exceed 0.87 at all overlaps and peak at 0.94 for overlap 0.4. The `DropFrom = Eve` configuration shows more fluctuation, with F1-scores ranging from 0.50 to 0.92, likely due to reduced training set size at lower overlaps.

Re-identification rates vary substantially across overlaps. In the `DropFrom = Both` setting, a peak re-identification rate of 6.85% is observed at overlap 0.6. Notably, this rate drops to near zero at 0.8, despite similar predictive metrics, indicating the effect of reduced unique reconstruction under high training similarity. The `DropFrom = Eve` setting shows similar behavior with a maximum re-identification rate of 6.12% at 0.6 overlap, largely driven by greedy matches.

The aggregate F1-to-re-identification relationship exhibits a clear positive trend, reinforcing that high structural accuracy translates into privacy loss at the record level. This is further supported by the distribution of fuzzy and greedy re-identifications, with both contributing significantly at intermediate overlaps.

Overall, the *euro person* dataset reveals that even realistic data encoded via Tabulation MinHash is susceptible to reconstruction and re-identification. The Dataset Extension Attack performs reliably across settings, with overlap 0.6 appearing to maximize re-identification success.

**Summary Across Datasets and Overlap Levels** Figure ?? in Appendix ?? illustrates the relationship between overlap and both re-identification rate (left) and F1-score (right), averaged over drop strategies for each dataset. The results highlight how both structural and record-level vulnerabilities evolve with dataset size and training coverage.

Larger datasets such as `fakename_20k`, `fakename_10k`, and `euro_person` exhibit clear monotonic trends. As the overlap increases, both F1-scores and re-identification rates improve. For `fakename_20k`, the re-identification rate peaks at nearly 12% for 0.8 overlap. Similarly, `euro_person` reaches its highest F1-score at 0.6 overlap, accompanied by its highest re-identification performance, indicating that structural accuracy enables effective exploitation.

Smaller datasets such as `fakename_1k`, `fakename_2k`, and `titanic_full` show limited gains in re-identification, regardless of overlap. Their F1-scores increase with overlap, but remain insufficient to enable consistent reconstruction. Notably, `fakename_5k` serves as a transition point: at 0.6 overlap, it achieves a modest re-identification rate and solid F1-score, but both metrics drop again at 0.8 overlap.

Across datasets, the plots reveal that higher overlap generally yields better F1-scores. However, effective re-identification is highly dependent on dataset size and complexity. The correlation between these two metrics is dataset-specific and non-linear, particularly at high F1 values where small increases may lead to steep jumps in re-identification.

These findings reinforce the conclusion that Tabulation MinHash encodings do not provide sufficient protection in high-overlap, large-scale scenarios. While the Dataset Extension Attack may remain ineffective on small or disjoint datasets, it becomes increasingly potent as structural

learning converges, ultimately compromising encoded identifiers.

**Architecture** Figure ?? in Appendix ?? presents a distributional analysis of the neural network architectures selected during hyperparameter optimization across all Tabulation MinHash experiments. The results provide insight into which architectural choices consistently yield strong performance in structural reconstruction and re-identification tasks.

The majority of optimized models employ a shallow architecture with a single hidden layer. More complex configurations with two or three layers are rare, suggesting that the Dataset Extension Attack can learn effective representations without requiring deep hierarchical abstraction. This is consistent with the relatively structured nature of the prediction task and the compactness of Tabulation MinHash encoded records.

In terms of hidden size, larger configurations are clearly favored. The most frequent choice is a hidden dimension of 2048, followed by 1024. Smaller sizes below 512 occur infrequently, indicating that wide layers contribute significantly to capturing the necessary n-gram co-occurrence patterns for decoding.

Dropout rates are broadly distributed, with a mean around 0.24. This suggests that regularization is helpful, but not critical; overfitting appears to be limited even with lower dropout. The threshold histogram indicates that models typically settle around a value of 0.44 for binary classification, though a wide range is explored, reflecting dataset-specific optimality.

Activation functions are led by `elu` and `selu`, both of which support smooth non-linear transitions and internal normalization. Simpler functions like `relu` and `gelu` are used less frequently. This preference for normalized activations may reflect the benefits of stability during training on small gradients.

Among optimizers, `AdamW` is most commonly selected, followed by `Adam` and `RMSprop`. This aligns with the need for stable adaptive learning rates and effective weight decay. Learning rate schedulers are also diverse, with `ReduceLROnPlateau` and `CyclicLR` being the most frequent. The inclusion of schedulers reflects their utility in adjusting learning dynamics over the short training windows.

Batch sizes of 8 and 16 dominate, likely due to the small to medium dataset sizes. Finally, the number of training epochs converges to 20 in nearly all cases, the maximum allowed during tuning, suggesting that performance continues to improve up to the epoch limit.

In summary, the optimal architecture for Tabulation MinHash encoded Dataset Extension Attack models is characterized by a shallow but wide feedforward network, mild regularization, normalized activations, and adaptive optimizers with scheduler support. These configurations are robust across datasets and contribute significantly to the attack's effectiveness.

### 4.3.2 Two-Step Hash

The following subsection focuses on the results obtained using the Two-Step Hash encoding scheme. In this analysis, each dataset is evaluated under varying settings. To ensure consistency, the dataset and encoding scheme are fixed, while the evaluation focuses on the impact of different overlap sizes and drop-from strategies.

**Titanic Full** On the `titanic_full` dataset, the Dataset Extension Attack performs reliably when using Two-Step Hash encoding, achieving stable precision and recall across overlap values (see Figure ?? in Appendix ??). F1-scores range from 0.34 to 0.83, depending on the drop strategy and overlap.

In the `DropFrom = Eve` setting, the model consistently achieves high precision (above 0.95) across all tested overlaps. Recall improves with overlap, rising from 0.56 at 0.7 to 0.74 at 0.9. The resulting F1-score increases accordingly, reaching a maximum of 0.83 at 0.9 overlap, indicating robust learning from partially re-identified data even in small datasets.

The `DropFrom = Both` configuration shows more variability. At 0.7 overlap, the F1-score remains low at 0.34, reflecting low precision (0.26) despite moderate recall (0.55). However, at 0.8 overlap, both precision and recall improve, yielding an F1-score of 0.72. This suggests that the `DropFrom = Both` strategy can be mitigated as overlap increases, but model reliability depends strongly on sample size.

No re-identifications are observed at any overlap level for either drop-from strategy. Both fuzzy and greedy matching yield zero hits, highlighting that accurate structural predictions do not necessarily lead to successful record reconstruction in small datasets like `titanic_full`.

In summary, Two-Step Hash provides solid predictive performance on the `titanic_full` dataset, particularly when Eve's overlap is high. Nonetheless, the attack remains structurally bounded, and record-level re-identification is not achieved.

**Fakename 1k** On the `fakename_1k` dataset, the Dataset Extension Attack shows measurable improvements in structural prediction performance as the overlap increases, though it fails to achieve any successful re-identifications (see Figure **??** in Appendix **??**).

In the `DropFrom = Eve` setting, F1-scores rise sharply with overlap, starting from 0.11 at 0.4 to 0.74 at 0.8. This trend is driven by improvements in both recall (from 0.06 to 0.62) and precision (from 0.52 to 0.95). The results suggest that the Dataset Extension Attack is able to learn useful representations even with a small dataset, provided sufficient re-identified training data is available.

The `DropFrom = Both` configuration performs worse, particularly at lower overlaps. At 0.6 overlap, F1-score remains low (0.07), but rises to 0.66 at 0.8, supported by high precision (0.95) and moderate recall (0.52).

Despite achieving F1-scores over 0.7 in some configurations, re-identification remains at 0% for all tested overlaps and matching strategies. This underlines the fact that a strong n-gram prediction model does not necessarily translate into record-level success when the dataset size is small.

Overall, Two-Step Hash yields competitive structural predictions on small datasets like `fakename_1k`, particularly when the training overlap is high. However, the Dataset Extension Attack does not succeed in reconstructing any individual records in this configuration.

**Fakename 2k** The `fakename_2k` dataset reveals a clear improvement in Dataset Extension Attack performance with increasing overlap when Two-Step Hash encoding is used (see Figure **??** in Appendix **??**). While low-overlap configurations perform poorly, both structural prediction and re-identification become feasible at higher overlaps.

In the `DropFrom = Eve` setting, the F1-score increases from 0.11 at 0.2 to 0.76 at 0.6, and further to 0.83 at 0.8. Precision remains high throughout (above 0.79), while recall improves substantially with overlap. These results indicate that once a sufficient number of re-identified records are available, the Dataset Extension Attack can learn and generalize the underlying encoding structure effectively.

The `DropFrom = Both` configuration follows a similar trend but lags slightly. At 0.4 overlap, performance remains negligible (F1 = 0.11), but at 0.8 it rises to 0.83, with both precision

(0.97) and recall (0.74) being strong. This confirms that the `DropFrom = Both` strategy can be overcome given sufficient overlap.

Unlike on the 1k dataset, the Dataset Extension Attack successfully re-identifies individual records at higher overlaps. Under `DropFrom = Eve`, a small re-identification rate of 0.1% is achieved at 0.6. In the `DropFrom = Both` setting, this increases to 0.3% at 0.8. Although the rates are modest, they demonstrate that re-identification becomes possible as structural accuracy increases.

The trend between F1-score and re-identification rate is consistent, with successful re-identifications only occurring beyond F1 = 0.75. This reflects a threshold-like behavior in the attack's effectiveness.

In summary, the `fakename_2k` dataset shows that Two-Step Hash encoding provides limited protection once the overlap exceeds 0.6. The Dataset Extension Attack can achieve strong predictions and initial re-identifications even in modestly sized datasets.

**Fakename 5k** The `fakename_5k` dataset marks the transition point where the Dataset Extension Attack begins to yield meaningful re-identification rates under Two-Step Hash encoding (see Figure **??** in Appendix **??**). Both precision and recall improve steadily with increasing overlap, and re-identification becomes feasible.

In the `DropFrom = Eve` setting, the model achieves an F1-score of 0.10 at 0.2 overlap, but quickly improves to 0.89 at 0.6, with near-perfect precision (0.99) and strong recall (0.82). Interestingly, performance slightly drops at 0.8, suggesting that the increased overlap does not always translate to higher generalization capacity under this drop strategy.

The `DropFrom = Both` configuration follows a smoother trajectory. F1-score increases from 0.22 at 0.2 overlap to 0.76 at 0.4, 0.89 at 0.6, and peaks at 0.94 at 0.8. These results are underpinned by consistently high precision and rising recall, indicating the model's ability to capture Tabulation MinHash encoded structure even under the `DropFrom = Both` strategy.

The first substantial re-identification rates are observed in this dataset. Under `DropFrom = Eve`, the combined rate reaches 2.1% at 0.6 but drops again at 0.8. In contrast, `DropFrom = Both` shows increasing re-identification, reaching 5.52% at 0.8.

The re-identification curve as a function of F1-score shows the expected upward trend, confirming that strong structural prediction is a prerequisite for record-level re-identification.

Overall, the `fakename_5k` dataset demonstrates that Dataset Extension Attack success under Two-Step Hash encoding is strongly dependent on overlap and dataset size. Once a critical threshold is reached, even structurally obfuscated data becomes vulnerable to reconstruction.

**Fakename 10k** The Dataset Extension Attack achieves consistently high structural prediction performance on the `fakename_10k` dataset when using Two-Step Hash encoding (see Figure **??** in Appendix **??**). Re-identification becomes substantial, particularly in the `DropFrom = Eve` configuration at intermediate overlaps.

In the `DropFrom = Eve` scenario, F1-scores are strong across all overlaps: 0.87 at 0.2, rising to 0.94 at 0.6, before slightly decreasing to 0.75 at 0.8. These fluctuations are primarily driven by changes in recall, which peaks at 0.90 but drops sharply at 0.8, despite stable precision above 0.96 throughout. This indicates that excessive overlap may not always improve performance due to overfitting or reduced variance in the training set.

The `DropFrom = Both` configuration shows more stability at high overlaps. F1-scores increase from 0.76 at 0.2 to 0.91 at 0.4, and finally 0.94 at 0.8. At 0.6, however, the model fails

to generalize, with recall collapsing to 0.05 and F1 dropping to 0.10. This suggests a sensitivity to training set composition at this overlap.

Re-identification rates are highest for Eve at 0.6 overlap (10.9%), coinciding with its peak in F1-score. The `DropFrom = Both` setup reaches a maximum re-identification rate of 5.6% at 0.8. Fuzzy and greedy matching contribute similarly to the results, with combined predictions amplifying the attack's effectiveness.

The re-identification rate shows a non-linear but positively correlated relationship with F1-score. The strongest privacy breaches occur in the range F1 $> 0.9$, highlighting the tipping point at which structural reconstruction translates into record-level re-identification.

In summary, the fakename 10k dataset confirms that Dataset Extension Attack success under Two-Step Hash encoding scales with dataset size and overlap. However, unexpected performance drops can occur under certain conditions, emphasizing the need for robust model selection.

**Fakename 20k** The `fakename_20k` dataset confirms the scalability and effectiveness of the Dataset Extension Attack when applied to large, Two-Step Hash encoded datasets (see Figure **??** in Appendix **??**).

In the `DropFrom = Eve` configuration, the Dataset Extension Attack achieves F1-scores of 0.94, 0.97, and 0.99 at overlaps 0.2, 0.4, and 0.8, respectively. Both precision and recall are near 1.0 throughout, indicating highly reliable reconstruction of the encoded representations. In the `DropFrom = Both` case, the model achieves an F1-score of 0.92 at 0.8 overlap with a precision of 0.99 and recall of 0.86, again showing strong generalization under `DropFrom = Both`.

Under `DropFrom = Eve`, the rate increases from 5.73% at 0.2 overlap to 13.20% at 0.4, and further to 23.00% at 0.8. The combined matching strategy benefits from contributions by both fuzzy and greedy methods, peaking at over 23%.

In the `DropFrom = Both` setting, the re-identification rate reaches 2.49% at 0.8 overlap. While lower than in the `DropFrom = Eve` case, this still represents a notable privacy breach.

The positive linear trend between F1-score and re-identification rate is especially clear in this dataset, with a nearly perfect fit across configurations. This reinforces the conclusion that once structural accuracy exceeds 0.9, re-identification becomes a likely outcome.

In summary, the `fakename_20k` experiments demonstrate high vulnerability of Two-Step Hash encoded databases under Dataset Extension Attacks. Even with minimal overlap, the Dataset Extension Attack can achieve both high reconstruction fidelity and impactful re-identification, posing a serious threat to encoded privacy-preserving record linkage.

**Europerson** The `euro_person` dataset yields the highest observed re-identification rates under Two-Step Hash encoding. Structural prediction performance is nearly perfect across all overlap values, and re-identification success exceeds 20% at high overlap.

In the `DropFrom = Eve` configuration, the model achieves F1-scores above 0.98 for overlaps 0.6 and 0.8. Precision and recall are both near 1.0, indicating that the model captures the encoded structure with high fidelity. Even at 0.2 overlap, the F1-score remains strong at 0.93, supported by a recall of 0.90 and precision of 0.95.

The `DropFrom = Both` setting shows similar behavior. The F1-score rises from 0.68 at 0.2 to 0.98 at 0.6 and 0.99 at 0.8. This suggests that sufficient overlap renders the Dataset Extension Attack robust even under this drop-from strategy.

Re-identification rates are particularly striking. Under `DropFrom = Eve`, the combined matching method reaches 22.6% at 0.6 overlap and remains stable at 22.3% at 0.8. The corresponding values in the `DropFrom = Both` setting are 17.8% and 21.9%, respectively.

The re-identification curve versus F1-score shows a near-linear relationship with a steep slope, underscoring the strong coupling between model accuracy and privacy leakage.

In summary, the `euro_person` dataset illustrates that the combination of Two-Step Hash encoding and realistic, high-quality input data is particularly vulnerable to Dataset Extension Attacks. The Dataset Extension Attack can achieve both high prediction fidelity and extensive re-identification success, even with limited initial overlap.

**Summary Across Datasets and Overlap Levels** The summary plots in Figure **??** in Appendix **??** illustrate the overall relationship between overlap, structural prediction quality, and re-identification success across all evaluated datasets using Two-Step Hash encoding.

Across all datasets, F1-scores generally increase with overlap, reflecting improved structural learning by the neural model as more training data becomes available. For small datasets like `titanic_full` and `fakename_1k`, the Dataset Extension Attack struggles until the overlap reaches 0.8. In contrast, medium-sized datasets such as `fakename_5k` and `fakename_10k` show steep improvements between 0.4 and 0.6. Larger datasets (`fakename_20k` and `euro_person`) achieve F1-scores above 0.9 even at lower overlaps (0.2 or 0.4), confirming their higher learnability.

Re-identification success remains low in small datasets, regardless of F1-score. In `fakename_1k`, `fakename_2k`, and `titanic_full`, the Dataset Extension Attack fails to re-identify any records. The first meaningful re-identification rates emerge in `fakename_5k`, reaching up to 2.5% at 0.8 overlap. In `fakename_10k`, the rate peaks at 5.8% at overlap 0.6 but decreases slightly at 0.8. `fakename_20k` shows relatively stable re-identification performance across overlaps, with rates of 5.7%, 13.2%, and 12.8% at overlaps 0.2, 0.4, and 0.8, respectively. The `euro_person` dataset is the most vulnerable, with a re-identification rate of 20.4% at 0.6 and peaking at 22.9% at 0.8.

These results confirm a non-linear but consistent dependency between structural performance and re-identification capability. Notably, some datasets exhibit diminishing returns at very high overlaps, potentially due to reduced variability or model overfitting. Additionally, privacy leakage appears to depend not only on model performance (F1) but also on dataset characteristics such as name structure, token diversity, and size.

Overall, Two-Step Hash poses a clear privacy risk under Dataset Extension Attacks. While encoding obfuscates raw values, neural networks trained on auxiliary data can learn the structural mapping and enable re-identification—especially in realistic, large-scale datasets.

**Architecture** Figure **??** in Appendix **??** shows the distributions of the best-performing neural network configurations for models trained on Two-Step Hash encoded datasets. Most models are relatively shallow: the majority use a single hidden layer, indicating that deeper networks are not necessary to capture the structure of the encoding. At the same time, the preferred hidden size is large, with 2048 being the most frequent choice, suggesting that a wide representation is crucial for learning the mapping from encoded inputs to class labels.

In terms of activation functions, `gelu` and `selu` are most often selected, while traditional choices like `relu`, `tanh`, and `leaky_relu` appear less frequently. This preference points to the importance of smooth and differentiable non-linearities that preserve gradient flow. Optimizer

selection favors RMSprop, followed by Adam and AdamW. RMSprop's popularity may be due to its ability to handle sparse and noisy gradients effectively. Learning rate schedulers are used in most cases, with None and CyclicLR dominating. This reflects that both fixed and adaptive schedules can work well depending on dataset size and overlap.

Regularization is handled via dropout, with most selected values ranging between 0.15 and 0.35 (mean: 0.25), indicating a moderate level of regularization is helpful to avoid overfitting to the structure of the extended training data. The decision threshold for classification tends to cluster around 0.39. Most models train for around 15–20 epochs, with a noticeable concentration at the upper bound, suggesting that full training cycles are often required to reach convergence. Small batch sizes, particularly 8 and 16, dominate, likely due to their regularizing effect and suitability for noisy structural input.

### 4.3.3 Bloom Filter

The following subsection focuses on the results obtained using the Bloom Filter encoding scheme. In this analysis, each dataset is evaluated under varying settings. To ensure consistency, the dataset and encoding scheme are fixed, while the evaluation focuses on the impact of different overlap sizes and drop-from strategies.

**Titanic Full** On the titanic_full dataset, the Dataset Extension Attack demonstrates strong structural learning capabilities when using Bloom Filter encoding (see Figure ?? in Appendix ??).

At overlap 0.9, the model reaches an F1-score of 0.83 under the DropFrom = Eve strategy, indicating that the structural mapping between encoded and raw data can be learned to a high degree. With the DropFrom = Both strategy, the model performs slightly worse with an F1-score of 0.81. While the precision is consistently high in both cases (above 0.95), recall increases significantly with higher overlap and is consistently better when training on DropFrom = Eve.

Despite strong structural performance, no successful re-identifications were made, regardless of overlap or re-identification strategy. Both fuzzy and greedy matching yield zero hits, highlighting that accurate structural predictions do not necessarily lead to successful record reconstruction in small datasets like titanic_full.

In summary, Bloom Filter provides solid predictive performance on the titanic_full dataset, particularly when DropFrom = Eve is used. Nonetheless, the attack remains structurally bounded, and record-level re-identification is not achieved.

**Fakename 1k** On the fakename_1k dataset, the Dataset Extension Attack shows moderate performance when using Bloom Filter encoding (see Figure ?? in Appendix ??). Performance varies notably depending on the drop strategy.

Under the DropFrom = Eve setting, the model achieves a peak F1-score of 0.67 at overlap 0.8. Precision reaches over 0.9, while recall follows a similar trend, confirming that overlap and drop strategy significantly influence learnability.

The DropFrom = Both configuration shows consistently lower performance, with F1-scores never exceeding 0.20. Precision remains low for this strategy, indicating that the model struggles to generalize well under this drop-from strategy.

Despite the difference in structural performance, no re-identifications are made at any overlap or with any matching method. As before, the attack manages to recover structural patterns

when training data is suitable, but Bloom Filters at this scale appear to prevent any meaningful semantic reconstruction.

Overall, Bloom Filter yields competitive structural predictions on small datasets like `fakename_1k`, particularly when the training overlap is high. However, the Dataset Extension Attack does not succeed in reconstructing any individual records in this configuration.

**Fakename 2k** For the `fakename_2k` dataset, the Dataset Extension Attack shows clear improvement in predictive performance with increasing overlap when Bloom Filter encoding is used (see Figure **??** in Appendix **??**).

Under the `DropFrom = Eve` setting, the attack reaches an F1-score of 0.86 at overlap 0.8. This is supported by both precision and recall exceeding 0.8, indicating robust learning from partially re-identified data.

The `DropFrom = Both` configuration shows weaker performance, yielding an F1-score of 0.72 at the same overlap. This discrepancy mirrors the gap in recall and precision between the two conditions, with recall remaining below 0.7 for the `DropFrom = Both` strategy.

Unlike previous datasets, Bloom Filter encoding begins to show measurable vulnerability. The re-identification rate reaches 1.5% in the best case (overlap 0.8, `DropFrom = Eve`), with most successful re-identifications resulting from the fuzzy matching strategy.

The comparison between F1-score and re-identification rate shows a clear positive correlation, confirming that structural accuracy translates to successful record-level attacks.

In summary, the `fakename_2k` dataset shows that Bloom Filter encoding provides limited protection once the overlap exceeds 0.8. The Dataset Extension Attack can achieve strong predictions and initial re-identifications even in modestly sized datasets.

**Fakename 5k** The `fakename_5k` dataset exhibits a noticeable rise in vulnerability under Bloom Filter encoding (see Figure **??** in Appendix **??**).

At overlap 0.6, the model trained on `DropFrom = Eve` achieves an F1-score of 0.93, clearly surpassing the performance of the model trained with the `DropFrom = Both` strategy (F1 = 0.90). For overlap 0.8, the scores diverge more strongly: the F1-score drops to 0.49 for `DropFrom = Eve`, while it remains high at 0.87 for `DropFrom = Both`. This inversion suggests a shift in learning dynamics where a larger overlap allows the model trained on `DropFrom = Both` to generalize better, possibly due to higher variance in the training set.

The re-identification rate peaks at overlap 0.6, with 5.1% when trained on `DropFrom = Eve` and 1.6% for `DropFrom = Both`.

Compared to `fakename_2k`, the attacker gains more leverage here. While the F1-score improves only slightly, the re-identification rate scales more than threefold, indicating that higher dataset contributes to exploitability.

The re-identification curve as a function of F1-score shows the expected upward trend, confirming that strong structural prediction is a prerequisite for record-level re-identification.

Overall, the `fakename_5k` dataset demonstrates that Dataset Extension Attack success under Bloom Filter encoding is strongly dependent on overlap and dataset size. Once a critical threshold is reached, even structurally obfuscated data becomes vulnerable to reconstruction.

**Fakename 10k** The results on `fakename_10k` mark a transition where the dataset size and model capacity jointly enable highly effective re-identification under Bloom Filter encoding (see Figure **??** in Appendix **??**).

The F1-score reaches a peak of 0.95 for `DropFrom = Eve` and 0.94 for `DropFrom = Both` at overlap 0.6. For both drop-from strategies, this overlap corresponds to the best-performing configuration. At overlap 0.8, both models maintain high performance, although `DropFrom = Both` starts to degrade more steeply (F1 = 0.81 vs. 0.88 for `DropFrom = Eve`).

Re-identification is most successful at overlap 0.6, with rates of 11.1% and 7.9% for `DropFrom = Eve` and `DropFrom = Both`, respectively. As before, the greedy matchins strategy yields the best results.

The fitted curve shows a positive relation between F1 and re-identification rate, confirming that performance gains increasingly translate to effective attacks beyond a certain threshold.

In summary, the `fakename_10k` dataset confirms that Dataset Extension Attack success under Bloom Filter encoding scales with dataset size and overlap. However, unexpected performance drops can occur under certain dropout conditions, emphasizing the need for robust model selection.

**Fakename 20k** The `fakename_20k` dataset yields high re-identification rates under Bloom Filter encoding (see Figure **??** in Appendix **??**).

At overlap 0.8, the trained model achieves F1-scores of 0.97 and 0.96 for `DropFrom = Eve` and `DropFrom = Both`, respectively. These near-perfect values are accompanied by precision values above 0.99 and recall values exceeding 0.93. The overall performance of the neural network remains stable across the entire overlap range, starting already at an F1-score of 0.83 for the lowest evaluated overlap of 0.2 (`DropFrom = Eve`).

Re-identification rates increase monotonically with overlap. At overlap 0.8, `DropFrom = Eve` yields a re-identification rate of 19.2%. `DropFrom = Both` yields a lower rate of 11.1%, but even this value is poses a serious threat. Decomposing the re-identification results reveals that the greedy method dominates.

The positive linear trend between F1-score and re-identification rate is especially clear in this dataset, with a nearly perfect fit across configurations. This reinforces the conclusion that once structural accuracy exceeds 0.9, re-identification becomes a likely outcome.

In summary, the `fakename_20k` experiments demonstrate high vulnerability of Bloom Filter encoded databases under Dataset Extension Attacks. Even with minimal overlap, the Dataset Extension Attack can achieve both high reconstruction fidelity and impactful re-identification, posing a serious threat to encoded privacy-preserving record linkage.

**Euro Person** The `euro_person` dataset yields the highest observed re-identification rates under Bloom Filter encoding (see Figure **??** in Appendix **??**).

Due to computational limitations encountered during the Graph Matching Attack phase, results are only available for overlap 0.8. The Graph Matching Attack failed to converge at lower overlap levels (0.2, 0.4, 0.6) due to the presence of non-finite values in the similarity matrix, which caused the singular value decomposition algorithm to encounter numerical instabilities. This limitation reflects the computational challenges associated with large-scale similarity computations in high-dimensional encoding spaces.

At overlap 0.8, the trained model achieves F1-scores of 0.9915 and 0.9886 for `DropFrom = Eve` and `DropFrom = Both`, respectively. These values are accompanied by near-perfect precision (0.9946 and 0.9961) and recall (0.9890 and 0.9819), underscoring the Dataset Extension Attack's capacity to learn the structural patterns of Bloom Filter encodings with high fidelity.

Re-identification rates reflect this structural success. `DropFrom = Eve` yields a re-identification rate of 28.75%, the highest recorded in the evaluation. `DropFrom = Both` results in a slightly lower but still critical rate of 20.77%. These results confirm that Bloom Filter encodings of semi-realistic identifiers become highly vulnerable to inference once a sufficient portion of the dataset is re-identified.

In summary, the `euro_person` experiments illustrate that Bloom Filter encodings fail to provide adequate protection under Dataset Extension Attacks in high-overlap, high-structure scenarios. The attack achieves both high predictive performance and substantial re-identification rates, confirming the practical threat to encoded privacy-preserving record linkage in real-world settings.

**Summary Across Datasets and Overlap Levels** Figure ?? in Appendix ?? illustrates the relationship between overlap and both re-identification rate (left) and F1-score (right), averaged over drop strategies for each dataset. The results highlight how both structural and record-level vulnerabilities evolve with dataset size and training coverage.

The overall performance of the Dataset Extension Attack with Bloom Filter encodings shows clear dependencies on both dataset size and overlap. Larger datasets enable significantly higher re-identification rates and F1-scores. On the `fakename_20k` dataset, re-identification peaks at 17.6% at an overlap of 0.6, while the F1-score consistently exceeds 0.95 across all overlap levels. Similarly, the `euro_person` dataset shows the highest observed re-identification rate of 28.75% at overlap 0.8, with a corresponding F1-score of 0.99.

In contrast, datasets with fewer than 5,000 entities yield only minimal re-identifications, even when model performance is moderately high. For instance, `fakename_2k` and `fakename_1k` exhibit low re-identification rates (below 1.5%), despite achieving F1-scores around 0.79 and 0.44, respectively, at high overlap. On `titanic_full`, the Dataset Extension Attack achieves modest F1-scores near 0.43 but fails to re-identify any records.

A general trend is observable: for all datasets, F1-score improves with increasing overlap. However, re-identification rates do not scale linearly with model performance. Some configurations (e.g., `fakename_10k` and `fakename_20k`) exhibit a peak at intermediate overlaps (e.g., 0.6), suggesting a trade-off between memorization and generalization during training. In these cases, higher overlap may reduce variance in the training set and unintentionally impair generalization.

These findings confirm that Bloom Filters remain a robust encoding for small-scale linkage scenarios but fail to offer sufficient protection in larger datasets under realistic threat models. The Dataset Extension Attack becomes increasingly effective with scale and overlap, particularly in the presence of sufficient training data and favorable Graph Matching Attack conditions.

**Architecture** Figure ?? in Appendix ?? presents a distributional analysis of the neural network architectures selected during hyperparameter optimization across all Bloom Filter experiments. The results provide insight into which architectural choices consistently yield strong performance in structural reconstruction and re-identification tasks.

The architectural configurations chosen during hyperparameter optimization for Bloom Filter encodings reveal a strong preference for compact, shallow networks. The majority of trained models used a single hidden layer, combined with a hidden size of 2048 neurons. Smaller hidden sizes were rarely selected.

Dropout rates clustered around 0.27 on average, with moderate variance, suggesting that regularization was important to avoid overfitting. Threshold values after sigmoid activation centered around 0.39, reflecting the relatively conservative decision boundaries necessary when working with noisy encodings.

Activation functions were balanced between `tanh`, `elu`, and `silu`, indicating no clear dominance. However, `RMSprop` was the most frequently chosen optimizer, reinforcing its robustness in non-convex, sparse-feature learning. Learning rate schedulers skewed toward `CyclicLR`, while the most frequent batch size was 8.

Training typically converged before or at 20 epochs, with a mean of approximately 16. These trends indicate that while Bloom Filter decoding requires large intermediate representations, the training process itself remains stable and efficient, even under early stopping.

In summary, the optimal architecture for Bloom Filter encoded Dataset Extension Attack models is characterized by a shallow but wide feedforward network, moderate regularization, balanced activations, and `RMSprop` optimization with cyclic learning rate scheduling. These configurations are robust across datasets and contribute significantly to the attack's effectiveness.

### 4.3.4 Comparison of Encoding Schemes

Figure **??** in Appendix **??** presents two line charts comparing the performance of the Dataset Extension Attack across three encoding schemes: Bloom Filter, Tabulation MinHash, and Two-Step Hash. The $x$-axis represents the dataset overlap between the encoded and auxiliary databases, while the $y$-axes report the re-identification rate (left plot) and F1-score (right plot), respectively. Each line corresponds to one encoding scheme, enabling a direct comparison of Dataset Extension Attack effectiveness on different encodings.

**F1-score.** The right graph reflects the model's structural reconstruction capability in terms of F1-score. All encoding schemes exhibit monotonic improvement with increasing overlap. Bloom Filter begins with a low F1-score (around 0.38) at low overlaps, then shows a steep rise to over 0.72 at 0.8. Tabulation MinHash improves steadily across the overlap range, peaking at about 0.67. Two-Step Hash shows the strongest reconstruction performance, achieving the highest F1-score of approximately 0.86 at overlap 0.8.

**Re-identification Rate.** The left graph shows that the re-identification rate increases with overlap for all encoding schemes up to an overlap of 0.6, with a continued rise or plateau at 0.8. This trend aligns with the intuition that more overlap improves the training signal for the neural model, enhancing its inference capability. Among the encodings, Two-Step Hash reaches the highest re-identification rate of approximately 5.9% at overlap 0.8. Bloom Filter follows closely, slightly surpassing Two-Step Hash at the highest overlap with a re-identification rate of around 6.1%. Tabulation MinHash consistently exhibits the lowest vulnerability, staying below 2.5% even at maximum overlap.

The comparison highlights Two-Step Hash as the most susceptible encoding under Dataset Extension Attack conditions, combining high structural learnability with re-identification risk. Bloom Filter presents moderate but growing vulnerability as overlap increases, while Tabulation MinHash offers the strongest resistance, characterized by both lower model performance and reduced re-identification success.

For some encodings, values for overlaps could be missing in some configurations (e.g., Bloom Filter in `euro_person`), which could skew the aggregate curves slightly. Nevertheless, the overall trend remains valid: Tabulation MinHash is the most secure encoding, and Bloom Filter and Two-Step Hash are both vulnerable to re-identification.

Figure **??** in Appendix **??** summarizes the average Dataset Extension Attack performance across datasets for each encoding scheme, comparing the F1/Dice scores aggregated over all evaluated overlap levels. Results are shown separately for both drop strategies (`DropFrom = Eve` and `DropFrom = Both`).

Across most datasets, Two-Step Hash consistently achieves the highest structural accuracy. It performs particularly well on `fakename_5k`, `fakename_10k`, and `titanic_full`, and maintains strong results across all drop strategies and dataset sizes. In `euro_person` and `fakename_20k`, the F1-scores are slightly lower than Bloom Filter, but remain close, confirming that Two-Step Hash enables expressive and generalizable structural learning.

Bloom Filter shows highly dataset-dependent performance. It performs best on large datasets like `fakename_20k` and `euro_person`, achieving near-perfect scores in both drop settings. However, its performance degrades sharply on smaller datasets like `fakename_1k` and `fakename_2k`, especially under the `DropFrom = Both` configuration. This reflects its high dependence on sufficient training data to yield usable reconstructions under the Dataset Extension Attack.

Tabulation MinHash demonstrates moderate and stable performance across all datasets. While it rarely achieves the top F1-scores, it avoids the severe drop-offs seen in Bloom Filter, indicating greater robustness to changes in dataset size and training quality. Its lower sensitivity to the drop strategy suggests reduced exploitability via dataset extension.

A consistent trend across encodings is that the `DropFrom = Eve` strategy yields better structural reconstruction than `DropFrom = Both`. This effect is especially pronounced on small and medium datasets (e.g., `fakename_1k`, `fakename_2k`, `fakename_5k`). For larger datasets such as `euro_person` and `fakename_20k`, the difference between the drop strategies becomes negligible, as the volume of training data compensates.

In summary, the bar chart analysis confirms that Two-Step Hash is structurally the most learnable encoding under the Dataset Extension Attack, while Tabulation MinHash offers more consistent resilience. Bloom Filter shows strong performance only under favorable training conditions, making it highly context-dependent in practice.

## 4.4 Discussion

### 4.4.1 Methodological Considerations and Setup Validity

The experimental design employed in this thesis was developed to evaluate the feasibility and effectiveness of Dataset Extension Attacks under controlled and reproducible conditions. The use of both synthetic (`fakename`) and semi-realistic (`euro_person`, `titanic_full`) datasets allows for a dual perspective: while `fakename` datasets provide consistency and tunability with respect to overlap and size, the `euro_person` dataset introduces characteristics of real-world name distributions, enhancing the external validity of the findings.

The Dataset Extension Attack pipeline was implemented modularly and applied uniformly across three encoding schemes: Bloom Filter, Tabulation MinHash, and Two-Step Hash, ensuring methodological consistency and enabling comparative analysis. Encoding specific preprocessing routines were explicitly separated from the training and reconstruction logic, avoiding model leakage or bias.

Differences in performance between the encoding schemes can, in part, be attributed to their structural characteristics. Tabulation MinHash encodings exhibit higher resistance to inference attacks due to their lower redundancy and increased internal randomness, while Two-Step Hash and Bloom Filter allow for more stable learning of structural patterns, particularly in large datasets.

Another methodological consideration lies in the dependence of the Dataset Extension Attack on the preceding Graph Matching Attack. Since the Dataset Extension Attack's training set is derived from the re-identification output of the Graph Matching Attack, the variability in Graph Matching Attack success rates, particularly on small or low-overlap datasets, has a direct effect on the Dataset Extension Attack's feasibility. In scenarios where the Graph Matching Attack fails to re-identify any individuals, no labeled data is available, rendering Dataset Extension Attack inapplicable. These constraints were respected throughout and such cases were excluded from the evaluation, as indicated in the results section.

Finally, all experiments were conducted on a standardized computing environment using uniform hyperparameter search space. This mitigates confounding due to hardware or stochastic optimization variance, although the evaluation still reflects the inherent non-determinism of neural network training to some degree.

### 4.4.2 Interpretation of Results

The results of the experiments confirm that the Dataset Extension Attack is capable of inferring sensitive information from encoded data, even when only partial ground truth is available through the Graph Matching Attack. Across all encoding schemes, the neural network successfully learned structural representations of the encodings and predicted n-gram distributions with high precision. This was particularly evident in larger datasets such as `fakename_20k` and `euro_person`, where F1-scores exceeded 0.95 and re-identification rates surpassed 20% under favorable conditions.

An important observation is that the Dataset Extension Attack retains utility even when the Graph Matching Attack fails to produce complete mappings. Partial re-identifications are sufficient for generating labeled training data, enabling the model to generalize to unmapped records. This demonstrates that the privacy risks posed by the Dataset Extension Attack extend beyond the intersection of datasets, directly challenging the assumption that encoded records are safe.

Another noteworthy outcome is the consistently high precision of the Dataset Extension Attack, often exceeding 0.95, even when recall remains modest. This behavior implies that while the model may fail to recover all true n-grams, the ones it does recover are usually correct. From a privacy perspective, this is a significant result: high precision enables human analysts or automated post-processing to narrow down candidate reconstructions with minimal noise. In practice, even partial reconstructions can be sufficient to compromise an individual's privacy, particularly when auxiliary data sources are available to complete the remaining information.

Furthermore, the experiments show that full reconstructions of plaintext identifiers are not a prerequisite for privacy compromise. In many cases, it is sufficient to correctly infer distinctive subsets of an identifier, such as rare name fragments or date-of-birth patterns. This weakens the effectiveness of encoding schemes that rely on obfuscating full strings while preserving local similarity for linkage. The Dataset Extension Attack exploits this design trade-off by targeting the underlying redundancy.

The relationship between structural reconstruction quality (F1-score) and re-identification

success is generally non-linear but positively correlated. The most effective attacks were observed when the F1-score exceeded 0.9, suggesting a threshold effect where marginal improvements in prediction quality lead to disproportionately large increases in re-identification capability. This threshold appears to vary depending on the dataset, overlap ratio, and the encoding scheme used.

An important nuance in interpreting the results lies in the relationship between F1-score and re-identification rate. Across many experiments, the Dataset Extension Attack achieved consistently high F1-scores, even when the re-identification rate remained low. This discrepancy suggests that the predicted n-gram sets were often structurally close to the true plaintext, with only a few missing or superfluous n-grams.

From a formal evaluation perspective, these small mismatches prevent a perfect match and thus lower the re-identification rate. However, from a human perspective, the amount of correctly predicted information may be sufficient to infer the full identifier. For example, if the Dataset Extension Attack predicts “He”, “nr”, “ic” for a surname, a human analyst could easily complete the reconstruction as “Henrich”, especially if assisted by contextual cues, frequency lists, or domain-specific expectations. This highlights a broader privacy risk: even imperfect reconstructions may provide enough semantic information to enable manual or semi-automated de-anonymization. Therefore, the re-identification rate may underestimate the practical compromise of privacy when high F1-scores are observed.

Overall, the experimental results support the hypothesis that learning-based inference attacks are a viable and effective threat to similarity-preserving Privacy-Preserving Record Linkage systems. The Dataset Extension Attack significantly expands the attack surface beyond what graph-based matching alone can achieve, enabling adversaries to perform probabilistic reconstructions at scale, even in the absence of direct overlaps.

### 4.4.3 Limitations and Practical Usefulness

While the Dataset Extension Attack demonstrates strong performance under controlled experimental conditions, several limitations must be acknowledged when interpreting the results and assessing their applicability to real-world scenarios.

First, the majority of experiments were conducted on synthetically generated datasets, particularly the `fakename` series. Although these datasets allow for systematic control over size, overlap, and naming structure, they lack the irregularities and inconsistencies often present in real data. Real-world datasets are likely to exhibit misspellings, incomplete fields, non-standard formats, and culturally diverse naming conventions. These factors introduce additional noise that could impair both the Graph Matching Attack and the Dataset Extension Attack, especially if n-gram distributions deviate significantly from the learned patterns.

Second, the overlap between the auxiliary and encoded datasets was varied in a controlled manner, with values ranging from 20% to 80%. In practice, such overlap ratios may not be precisely known or may be significantly lower. As the results indicate, the effectiveness of the Dataset Extension Attack degrades sharply in low-overlap scenarios due to insufficient labeled data for training. Although the attack retains some utility through its high precision, the recall drops substantially in such settings, limiting the attacker’s coverage.

Third, the scalability of the Dataset Extension Attack to large, heterogeneous databases remains a challenge. While the models trained on datasets with up to 20,000 entries showed strong performance, inference and reconstruction in real-world systems may involve millions of records. Although the pipeline was optimized for GPU acceleration and memory efficiency,

further engineering effort would be required to operate at web-scale, particularly during hyperparameter tuning and batch inference.

Another limitation lies in the decision to exclude the Large Language Model based reconstruction strategy from the main evaluation due to cost, latency, and reproducibility constraints. However, preliminary experiments suggest that Large Language Models could provide enhanced reconstruction quality, especially in ambiguous or sparse cases where graph- or dictionary-based strategies fail. Their integration into the attack pipeline represents a potential avenue for even more effective and human-like inference.

Despite these limitations, the Dataset Extension Attack remains a practically relevant threat. It can operate in semi-realistic environments, generalizes across encoding schemes, and maintains high interpretability. Even when full re-identification is not achieved, partial reconstructions can yield actionable information.

# 5 Conclusion

## 5.1 Summary

This thesis investigated the vulnerabilities in Privacy-Preserving Record Linkage systems with a particular focus on the feasibility and effectiveness of Dataset Extension Attacks. As the integration of sensitive data across institutional boundaries becomes increasingly important, particularly in sectors such as healthcare, finance, and public security, the need for secure Privacy-Preserving Record Linkage methods continues to grow. However, as this thesis has shown, widely used encoding techniques such as Bloom Filter, Tabulation MinHash, and Two-Step Hash remain vulnerable to Graph Matching Attacks and inference based re-identification methods like the Dataset Extension Attack.

The primary research question guiding this work was whether it is possible to extend the capabilities of the Graph Matching Attack to re-identify not only overlapping individuals between plaintext and encoded datasets, but also individuals outside the intersection set. To this end, the Dataset Extension Attack was proposed as a novel, learning-based extension of the Graph Matching Attack, capable of reconstructing plaintext information from encoded representations using Artificial Neural Networks trained on partially re-identified records. And from the results of the analysis it can be seen, that the Dataset Extension Attack is indeed capable of reconstructing a significant portion of the original identifiers, even when the overlap between plaintext and encoded datasets is limited. Also the Dataset Extension Attack outperforms baseline guessing methods most of the time, demonstrating the potential of learning-based inference attacks against Privacy-Preserving Record Linkage systems.

The key contributions of this thesis are threefold. First, a modular and extensible Dataset Extension Attack pipeline was developed, capable of adapting to different encoding schemes. Second, a tailored training and inference setup was implemented, leveraging PyTorch and GPU acceleration to efficiently train Artificial Neural Networks on n-gram prediction tasks. Third, a comprehensive experimental framework was established to benchmark the attack against multiple datasets, overlap levels, and encoding strategies, providing a foundation for comparative evaluation.

By building upon a refined Graph Matching Attack implementation, and extending it into a supervised inference-based attack, this work provides a concrete demonstration of how existing privacy preserving techniques may be overcome under realistic assumptions.

The Dataset Extension Attack introduced in this thesis follows a modular and systematic six-step pipeline designed to generalize across encoding schemes. Beginning with a Graph Matching Attack to establish a baseline set of re-identified individuals, the Dataset Extension Attack leverages these mappings as supervised training data for a neural model tasked with predicting n-gram structures in encoded representations.

To enable flexible experimentation, the attack pipeline was designed to accommodate three different encoding schemes: Bloom Filter, Tabulation MinHash, and Two-Step Hash. While the Graph Matching Attack itself is adaptable to the specific encoding, the Dataset Extension Attack requires encoding-specific preprocessing routines, Artificial Neural Network architectures,

and dataset representations. For instance, Bloom Filter and Tabulation MinHash produce fixed-length bit vectors, which are converted directly into tensors, whereas Two-Step Hash produces variable length integer sets that require normalization via one-hot encoding over a global dictionary of observed values.

A standardized data preparation process transforms the Graph Matching Attack output into structured datasets consisting of encoded inputs and corresponding multi-label n-gram labels. These datasets are then split into training, validation, and test subsets, with hyperparameter optimization performed via Ray Tune and Optuna using the Dice coefficient as the optimization target. This metric was chosen for its ability to balance precision and recall in multi-label classification tasks, and for its interpretability in the context of reconstructing partial identifiers.

Each experimental run varies the attacker's knowledge by changing the dataset overlap between the plaintext and encoded databases, simulating realistic levels of auxiliary data availability. For every setting, a dedicated neural network is trained using the re-identified individuals as labeled data, and inference is performed on the remaining unmapped encodings. To assess model performance, both baseline frequency-based guessing and two deterministic reconstruction strategies, graph-based and dictionary-based, are used for comparative evaluation.

The core of the Dataset Extension Attack's inference capability is a supervised Artificial Neural Network trained to predict the presence of character n-grams in encoded representations. This reframes the attack as a multi-label classification problem, where each output neuron corresponds to a possible n-gram, and the model predicts the probability of its inclusion in the plaintext. Despite the theoretical irreversibility of the underlying hash functions, the deterministic encoding structure allows the network to learn statistical associations across large training sets, thereby enabling probabilistic inference of plaintext components.

Nonetheless, the Dataset Extension Attack cannot achieve perfect reconstruction due to the inherent limitations of similarity-preserving encoding schemes, most notably hash collisions and the lossy nature of Bloom Filter. Instead, it operates under a probabilistic threat model, where the goal is to maximize partial reconstruction and re-identification success under practical constraints. The attacker is assumed to possess no cryptographic secrets or salts but may exploit knowledge of system design parameters and publicly available data, consistent with Kerckhoffs's principle and prior work on Graph Matching Attack based attacks.

To translate predicted n-grams into human interpretable identifiers, three reconstruction strategies were developed. The first method builds a directed graph from the predicted n-grams and attempts to find the longest coherent path through the character transitions. The second leverages auxiliary dictionaries of common first names, surnames, and birthdates to identify plausible matches via similarity scoring. The third strategy, based on Large Language Models, uses a prompt based generative approach to reconstruct full identifiers from the predicted tokens. While powerful, the latter was excluded from the main evaluation due to its high variability, external dependencies, and limited reproducibility.

Each reconstruction strategy represents a different level of attacker sophistication, from deterministic heuristics to semantic matching. Together, they illustrate the extent to which encoded identifiers can be reversed into personal data using only structural and statistical clues, even when access to encoding secrets is restricted.

The comprehensive experimental evaluation of the Dataset Extension Attack across multiple datasets, encoding schemes, and overlap configurations revealed significant vulnerabilities in widely deployed Privacy-Preserving Record Linkage systems. In total, 180 experiments were

conducted, systematically varying dataset sizes from 1,000 to 26,625 records, overlap ratios from 20% to 80%, and three different encoding schemes (Bloom Filter, Tabulation MinHash, and Two-Step Hash).

The results demonstrate that the Dataset Extension Attack consistently outperforms baseline frequency-based guessing methods, achieving an average re-identification rate of 2.95% across all experiments, with peak performance reaching 28.75% under optimal conditions. The attack's effectiveness scales strongly with both dataset size and overlap ratio, with larger datasets enabling more sophisticated pattern learning and higher overlaps providing richer training data.

Among the three encoding schemes evaluated, Two-Step Hash exhibited the highest vulnerability, achieving an average re-identification rate of 3.26% and the best structural learnability with an average F1-score of 0.688. This vulnerability stems from Two-Step Hash's deterministic encoding structure, which preserves more statistical regularities than the other schemes. The Bloom Filter encoding showed moderate vulnerability with an average re-identification rate of 3.8% and F1-score of 0.606, while Tabulation MinHash demonstrated the highest resilience with the lowest average re-identification rate (1.54%) and F1-score of 0.593.

The relationship between structural reconstruction quality (F1-score) and re-identification success revealed a threshold effect: when the F1-score exceeded 0.9, the Dataset Extension Attack achieved disproportionately higher re-identification rates. This threshold behavior was particularly evident in larger datasets such as `fakename_20k` and `euro_person`, where F1-scores above 0.95 consistently translated to re-identification rates exceeding 10%.

Dataset size emerged as a crucial factor in attack effectiveness. Small datasets (1,000-2,000 records) showed limited vulnerability, with re-identification rates typically below 1% even under high overlap conditions. However, medium-sized datasets (5,000-10,000 records) marked a transition point where the Dataset Extension Attack began achieving meaningful re-identification success, with rates ranging from 2% to 11%. Large datasets (20,000+ records) demonstrated the highest vulnerability, with re-identification rates reaching 13-28% under favorable conditions.

The overlap ratio between auxiliary and target datasets proved to an important parameter affecting attack success. Low overlap scenarios (20-40%) generally resulted in poor performance due to insufficient training data, while high overlap scenarios (60-80%) enabled the Dataset Extension Attack to achieve its maximum effectiveness.

The experimental results also revealed important differences between the two drop-from strategies employed. The `DropFrom = Eve` strategy, where the auxiliary dataset is a strict subset of the target dataset, generally yielded better performance than the `DropFrom = Both` strategy, particularly on smaller datasets. This difference diminished on larger datasets, where both strategies achieved comparable results.

Notably, the Dataset Extension Attack demonstrated high precision across all experiments, often exceeding 0.95, even when recall remained modest. This high precision is particularly concerning from a privacy perspective, as it means that the n-grams the model does predict are usually correct, enabling human analysts or automated systems to reconstruct partial identifiers with minimal noise.

The results also highlighted the practical feasibility of the attack: the Dataset Extension Attack pipeline operated efficiently on standard hardware, with hyperparameter optimization typically completing within reasonable time and inference scaling linearly with dataset size. This accessibility, combined with the attack's effectiveness, underscores the urgent need for more robust Privacy-Preserving Record Linkage mechanisms.

The findings and design of this thesis underscore an interesting insight: even without access to cryptographic secrets or exact encoding parameters, an attacker can exploit structural patterns and statistical regularities within Privacy-Preserving Record Linkage encoded data to perform effective re-identification. The Dataset Extension Attack demonstrates how partial re-identifications obtained via a Graph Matching Attack can serve as a springboard for broader inference, ultimately undermining the core privacy guarantees of non-interactive, similarity-preserving linkage protocols.

This threat is particularly acute in real-world deployments where auxiliary data sources are abundant and overlap with target datasets is likely. As demonstrated in the Dataset Extension Attack setup, even modest overlap rates can yield a meaningful training base for the neural model. Combined with efficient reconstruction strategies, this enables the attacker to recover a significant portion of the original dataset content, thereby transforming a partial breach into a systemic compromise.

The modularity of the Dataset Extension Attack pipeline allows it to be extended to new encoding schemes, auxiliary data sources, or target attributes with minimal adjustments. Moreover, the use of accessible deep learning frameworks and off-the-shelf hardware illustrates that the technical barrier for executing a Dataset Extension Attack is relatively low, further emphasizing the urgency for more robust Privacy-Preserving Record Linkage mechanisms.

## 5.2 Future Work

While the Dataset Extension Attack presented in this thesis establishes a practical and effective pipeline for reconstructing identifiers from encoded representations, several opportunities remain for future exploration, particularly in improving robustness under more challenging conditions. One of the central limitations of the current approach lies in its dependence on supervised learning, which requires labeled data produced by the Graph Matching Attack. In low-overlap scenarios, however, the number of successfully re-identified individuals becomes too small to support effective training, limiting the applicability of the Dataset Extension Attack. Future work could explore alternative approaches to increase the practical applicability of the Dataset Extension Attack, particularly in real-world settings where overlap between auxiliary and target datasets can be low.

Another line of research could involve reversing the modeling direction altogether. Instead of predicting plaintext n-grams from encoded representations, one could design a model that learns to predict the corresponding encoded representation from a given set of n-grams.

This would effectively invert the learning task, enabling a greedy reconstruction attack wherein an attacker iteratively constructs candidate plaintexts by adding one n-gram at a time. At each step, the model would generate a candidate encoding from the current n-gram set, and the attacker would retain the combination that maximizes similarity to the target encoded record. By fixing one n-gram and expanding the set greedily, the attacker could approximate the original encoding through iterative refinement, even in the absence of a direct mapping from encoding to n-grams.

In parallel, a reversed variant of the Dataset Extension Attack could be explored, in which the attacker does not attempt to infer plaintext from encoded data directly, but rather aims to test specific hypotheses about potential individuals. In this setting, the attacker begins with a curated list of plausible names or identifiers, either generated using an Large Language Model, derived from auxiliary datasets, or constructed via combinatorial enumeration. These

candidates are then re-encoded using the trained model, and their encodings are compared against the set of unreidentified records using similarity metrics. This enables targeted queries such as: "Is a particular individual part of the dataset?" or "Which of these candidate names best matches the encoded record?"

This model-informed dictionary attack can operate in both brute-force and guided modes. In the brute-force variant, a large number of combinations are tested exhaustively. In the guided variant, candidate generation and ranking are refined using probabilistic heuristics or learned scoring functions. If the attacker possesses partial knowledge of the input distribution, such as region-specific name frequency statistics, known first names, or institutional contexts, this reversed strategy may be particularly effective. It is especially relevant for encoding schemes that preserve structural locality or token frequency, such as Bloom Filter or Tabulation MinHash, where approximate matches can yield meaningful meaning even in the presence of collisions or noise.

Finally, future work could explore a more principled connection between the encoding schemes and the architecture of the neural model itself. Since encoding processes such as Bloom Filter or Tabulation MinHash can be described algebraically as matrix operations over hashed n-gram inputs, it may be possible to formalize the inverse mapping task as a linear or non-linear transformation. This formulation could inform the design of the Artificial Neural Network to allow for analytical derivation of network parameters, potentially reducing the reliance on extensive hyperparameter optimization. A deeper understanding of the encoding process as a differentiable or compositional function may also open the door to incorporating domain-specific inductive biases into the network, thereby improving learning efficiency and interpretability.

Beyond architectural enhancements, future work could also focus on algorithmic improvements to the Dataset Extension Attack pipeline itself. One direction could be to establish a tighter integration between the Graph Matching Attack and the Dataset Extension Attack, transforming the overall process into an iterative framework. In such a setup, the Dataset Extension Attack could be applied not as a one-shot inference step, but as a looped refinement mechanism that feeds its most confident predictions back into the Graph Matching Attack module. These new pseudo-labels could then expand the training set for the next Dataset Extension Attack iteration, gradually enlarging the set of re-identified individuals through a bootstrapping process. This would blend graph-based and learning-based re-identification strategies into a single adaptive attack.

Another important research direction is to systematically evaluate the robustness of the Dataset Extension Attack in the presence of privacy-enhancing defenses. While this thesis focused on attacking standard encoding schemes without countermeasures, real-world systems could incorporate protective mechanisms such as n-gram dropout, salting, or the injection of synthetic noise to reduce re-identification risk. More advanced approaches, such as applying differential privacy during encoding, or using diffusion-based variants of Bloom Filter, aim to further obfuscate the relationship between plaintext and encoded representations.

Complementary to algorithmic defenses, future work should also examine the generalization ability of the Dataset Extension Attack across a broader range of datasets. Current evaluations rely on synthetically generated or cleaned datasets with relatively homogenous, Western-centric name distributions. Expanding this scope to include real-world datasets with naturally occurring noise, misspellings, and inconsistencies would better simulate deployment conditions and test model robustness. Additionally, integrating multilingual datasets or culturally diverse naming conventions would challenge the assumptions of the current pipeline and

reconstruction strategies. For example, compound names, non-Latin scripts, or cultural variations in name structure may significantly impact n-gram frequency distributions and encoding patterns, potentially altering the effectiveness of both inference and reconstruction stages.

Another promising yet currently unexplored direction involves the integration of Large Language Models into the Dataset Extension Attack attack pipeline. Although excluded from the core evaluation due to reproducibility concerns and computational cost, Large Language Models offer a powerful foundation for semantic reconstruction strategies. Future work could investigate cost-effective approaches to Large Language Model integration, such as prompt tuning, knowledge distillation into smaller models, or constrained decoding methods tailored to the structure of Privacy-Preserving Record Linkage encoded data. These techniques may offer a balance between the expressive power of generative models and the practical constraints of attack scalability.

Finally, future research should also consider the broader impact and ethical implications of Dataset Extension Attack style attacks. While the primary aim of this thesis is to demonstrate the feasibility of inference-based attacks against Privacy-Preserving Record Linkage systems, these methods could also be repurposed as auditing tools for evaluating the real-world resilience of deployed Privacy-Preserving Record Linkage. In particular, regulatory bodies, data custodians, or third-party evaluators could apply controlled versions of the Dataset Extension Attack pipeline to assess whether deployed encodings are vulnerable to realistic adversarial inference under plausible auxiliary knowledge assumptions.

This naturally motivates a call for stronger privacy-preserving mechanisms and more standardized frameworks for risk evaluation in Privacy-Preserving Record Linkage deployments. Current encoding schemes often lack formal guarantees against inference attacks, especially when the encoded data must remain linkable and non-interactive. Future research could explore ways to strengthen existing Privacy-Preserving Record Linkage protocols through techniques such as probabilistic obfuscation, hybrid interactive schemes, or encoding structures designed to minimize semantic leakage. Ultimately, by highlighting concrete vulnerabilities and proposing paths toward mitigation, this line of work aims not only to critique existing systems, but also to contribute to the development of more robust and transparent Privacy-Preserving Record Linkage technologies.

# Bibliography


[AHS23] Frederik Armknecht, Youzhe Heng, and Rainer Schnell. "Strengthening privacy-preserving record linkage using diffusion." In: *Proceedings on Privacy Enhancing Technologies* (2023).

[BGF17] Facundo Bre, Juan Gimenez, and Víctor Fachinotti. "Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks." In: *Energy and Buildings* 158 (Nov. 2017). DOI: 10.1016/j.enbuild.2017.11.045.

[Blo70] Burton H Bloom. "Space/time trade-offs in hash coding with allowable errors." In: *Communications of the ACM* 13.7 (1970), pp. 422–426.

[Bro97] Andrei Z Broder. "On the resemblance and containment of documents." In: *Proceedings. Compression and Complexity of SEQUENCES 1997*. IEEE. 1997, pp. 21–29.

[DKK+12] AD Dongare, RR Kharde, Amit D Kachare, et al. "Introduction to artificial neural network." In: *International Journal of Engineering and Innovative Technology (IJEIT)* 2.1 (2012), pp. 189–194.

[FS69] Ivan P Fellegi and Alan B Sunter. "A theory for record linkage." In: *Journal of the American Statistical Association* 64.328 (1969), pp. 1183–1210.

[Fou] The Linux Foundation. *PyTorch — pytorch.org*. https://pytorch.org. [Accessed 12-03-2025].

[GB10] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[GSB+21] Fardin Ghorbani, Javad Shabanpour, Sina Beyraghi, Hossein Soleimani, Homayoon Oraizi, and M. Soleimani. "A deep learning approach for inverse design of the metasurface for dual-polarized waves." In: *Applied Physics A* 127 (Nov. 2021), p. 869. DOI: 10.1007/s00339-021-05030-6.

[HCR+16] Francisco Herrera, Francisco Charte, Antonio J Rivera, María J Del Jesus, Francisco Herrera, Francisco Charte, Antonio J Rivera, and María J del Jesus. *Multilabel classification*. Springer, 2016.

[HSW07] Thomas N Herzog, Fritz J Scheuren, and William E Winkler. *Data quality and record linkage techniques*. Vol. 1. Springer, 2007.

[IH18] Jim Isaak and Mina J Hanna. "User data privacy: Facebook, Cambridge Analytica, and privacy protection." In: *Computer* 51.8 (2018), pp. 56–59.

[KKM+14] Hye-Chung Kum, Ashok Krishnamurthy, Ashwin Machanavajjhala, Michael K Reiter, and Stanley Ahalt. "Privacy preserving interactive record linkage (PPIRL)." In: *Journal of the American Medical Informatics Association* 21.2 (2014), pp. 212–220.

[KM24] Jennifer King and Caroline Meinhardt. *Rethinking Privacy in the AI Era: Policy Provocations for a Data-Centric World*. White Paper, Stanford University Institute for Human-Centered Artificial Intelligence (HAI). 2024. URL: http://www.darkpatternstipline.org.

[MJ+01] Larry R Medsker, Lakhmi Jain, et al. "Recurrent neural networks." In: *Design and Applications* 5.64-67 (2001), p. 2.

[MK19] Karl Manheim and Lyric Kaplan. "Artificial intelligence: Risks to privacy and democracy." In: *Yale JL & Tech.* 21 (2019), p. 106.

[ON15] Keiron O'shea and Ryan Nash. "An introduction to convolutional neural networks." In: *arXiv preprint arXiv:1511.08458* (2015).

[PSZ+24] Aditi Pathak, Laina Serrer, Daniela Zapata, Raymond King, Lisa B Mirel, Thomas Sukalac, Arunkumar Srinivasan, Patrick Baier, Meera Bhalla, Corinne David-Ferdon, et al. "Privacy preserving record linkage for public health action: opportunities and challenges." In: *Journal of the American Medical Informatics Association* 31.11 (2024), pp. 2605–2612.

[RCS20] Thilina Ranbaduge, Peter Christen, and Rainer Schnell. "Secure and accurate two-step hash encoding for privacy-preserving record linkage." In: *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part II 24*. Springer. 2020, pp. 139–151.

[RN16] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. pearson, 2016.

[SAH24] Jochen Schäfer, Frederik Armknecht, and Youzhe Heng. "R+ R: Revisiting Graph Matching Attacks on Privacy-Preserving Record Linkage." In: *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE. 2024, pp. 699–715.

[SBR09] Rainer Schnell, Tobias Bachteler, and Jörg Reiher. "Privacy-preserving record linkage using Bloom filters." In: *BMC medical informatics and decision making* 9 (2009), pp. 1–11.

[SMNP24] Vivek S Sharma, Shubham Mahajan, Anand Nayyar, and Amit Kant Pandit. *Deep Learning in Engineering, Energy and Finance: Principals and Applications*. CRC Press, 2024.

[SSA17] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. "Activation functions in neural networks." In: *Towards Data Sci* 6.12 (2017), pp. 310–316.

[Smi16] Tanshanika T Smith. *Examining data privacy breaches in healthcare*. Walden University, 2016.

[Smi17] Duncan Smith. "Secure pseudonymisation for privacy-preserving probabilistic record linkage." In: *Journal of Information Security and Applications* 34 (2017), pp. 271–279.

[VCRS20] Anushka Vidanage, Peter Christen, Thilina Ranbaduge, and Rainer Schnell. "A graph matching attack on privacy-preserving record linkage." In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 1485–1494.

[VSCR17] Dinusha Vatsalan, Ziad Sehili, Peter Christen, and Erhard Rahm. "Privacy-preserving record linkage for big data: Current approaches and research challenges." In: *Handbook of big data technologies* (2017), pp. 851–895.

# A Auxiliary Information

## A.1 Tabulation MinHash: Dataset Extension Attack Results

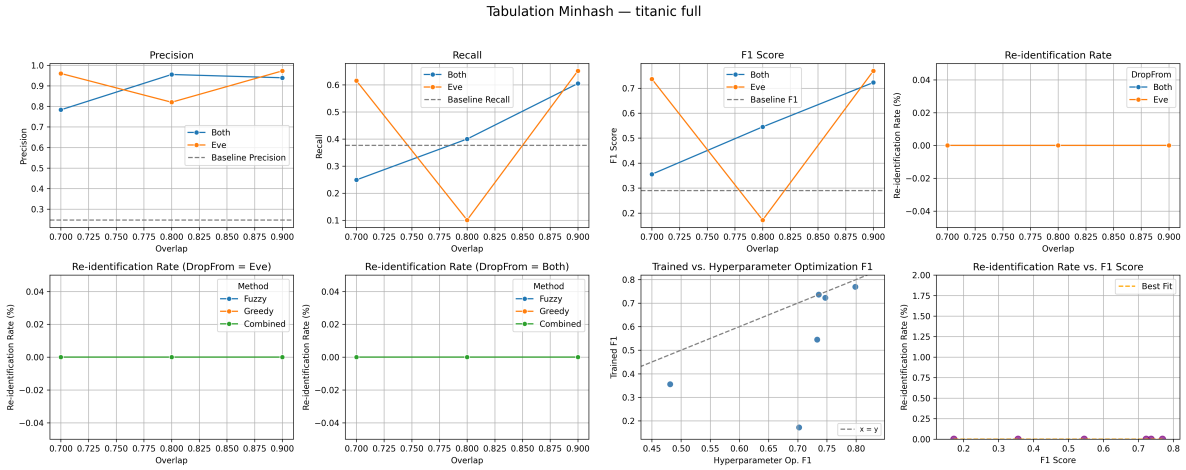


**Figure A.1:** Tabulation MinHash results on the `titanic_full` dataset.

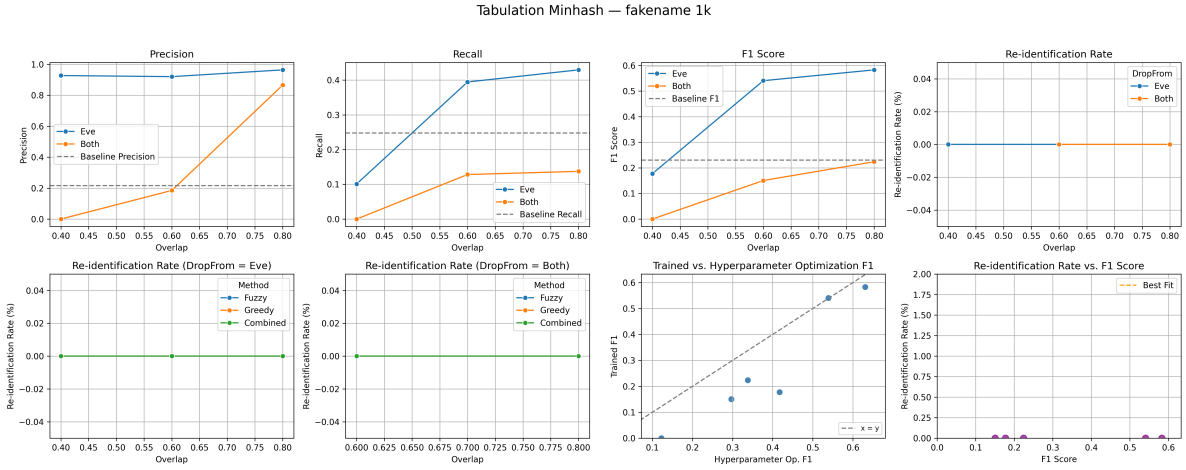


**Figure A.2:** Tabulation MinHash results on the `fakename_1k` dataset.

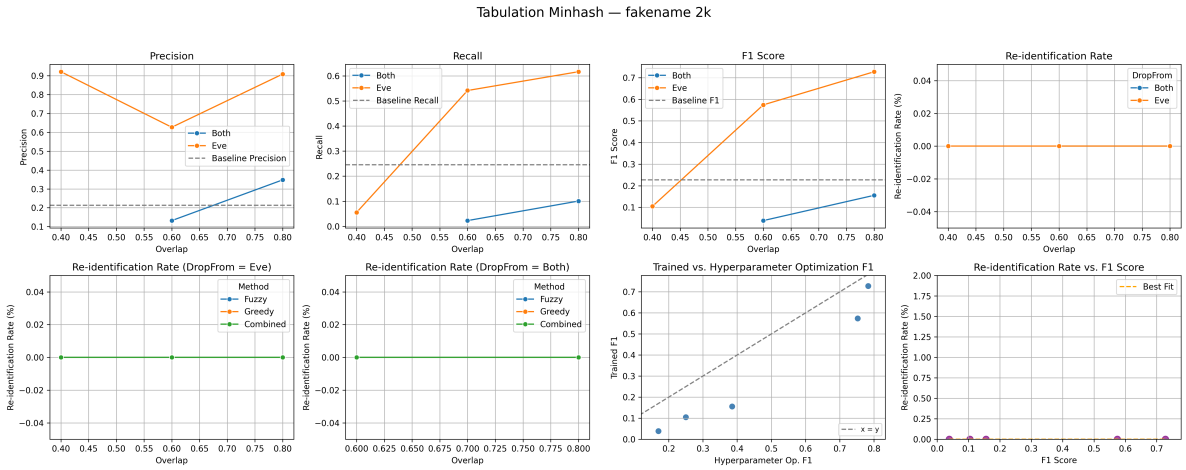


**Figure A.3:** Tabulation MinHash results on the `fakename_2k` dataset.

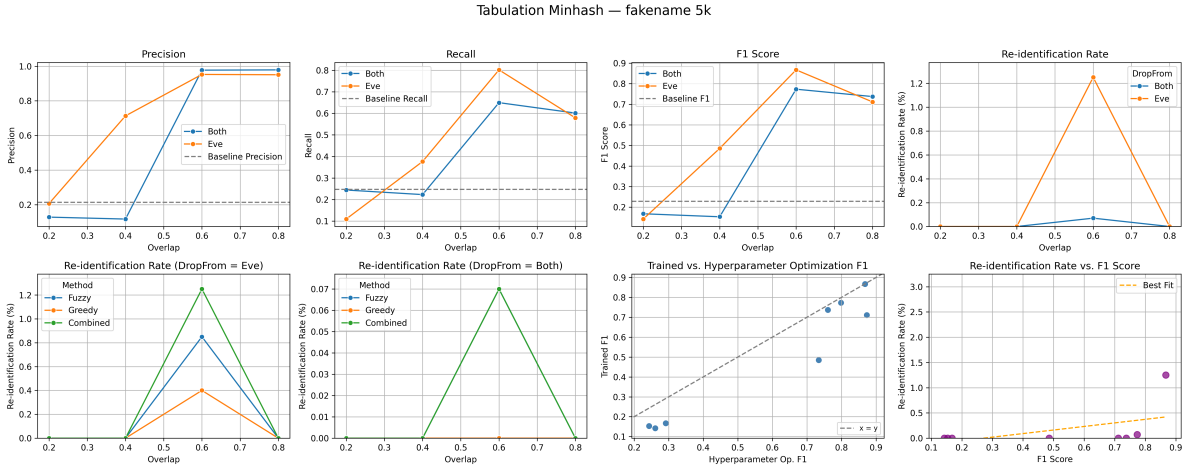


**Figure A.4:** Tabulation MinHash results on the `fakename_5k` dataset.

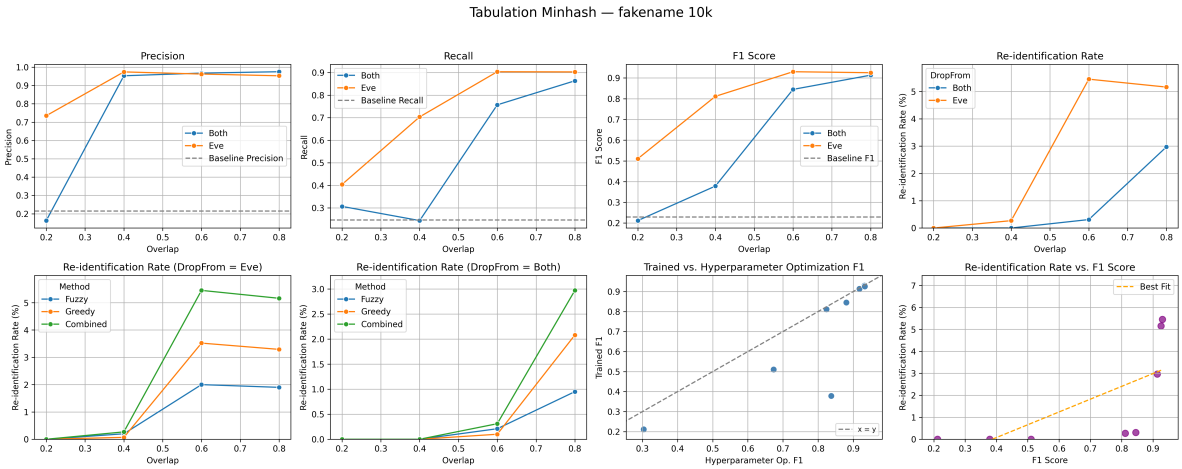


**Figure A.5:** Tabulation MinHash results on the `fakename_10k` dataset.

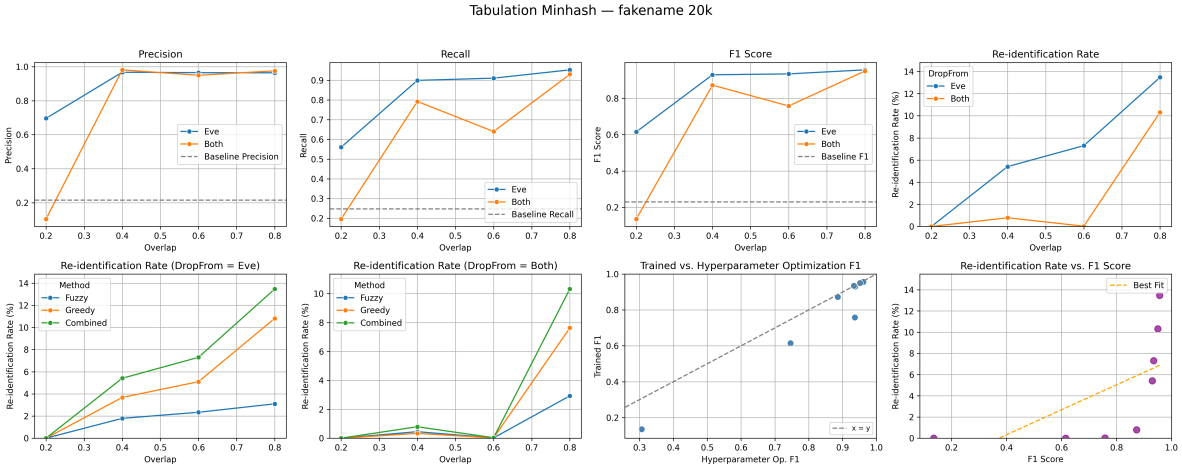


**Figure A.6:** Tabulation MinHash results on the `fakename_20k` dataset.

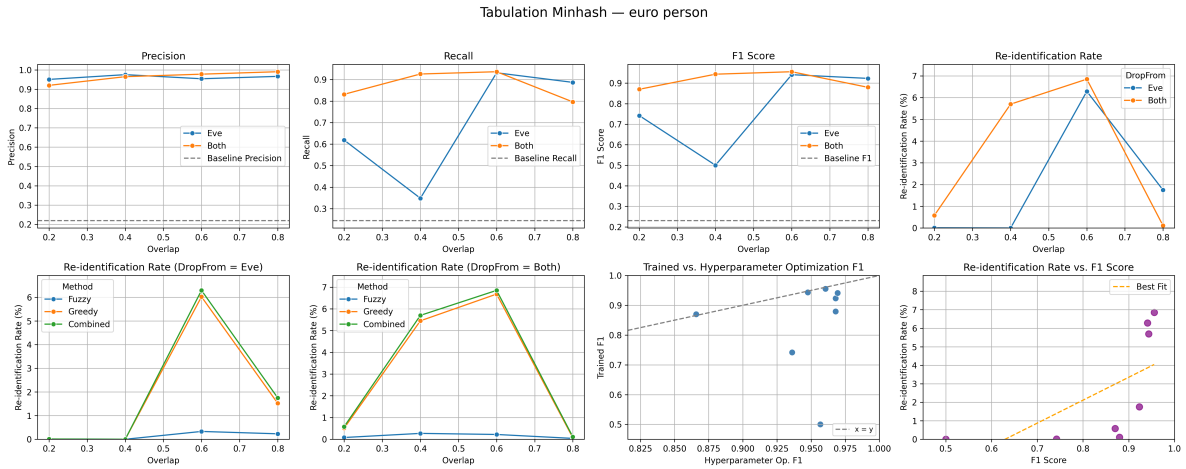


**Figure A.7:** Tabulation MinHash results on the `euro_person` dataset.

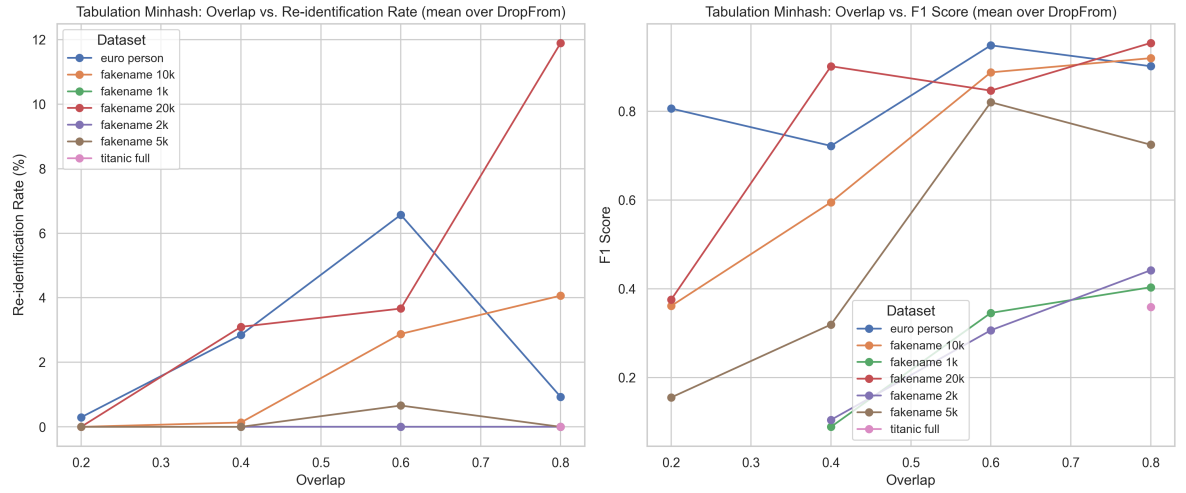


**Figure A.8:** Comparison of re-identification rates and F1 scores across all datasets with Tabulation MinHash encoding as a function of overlap.

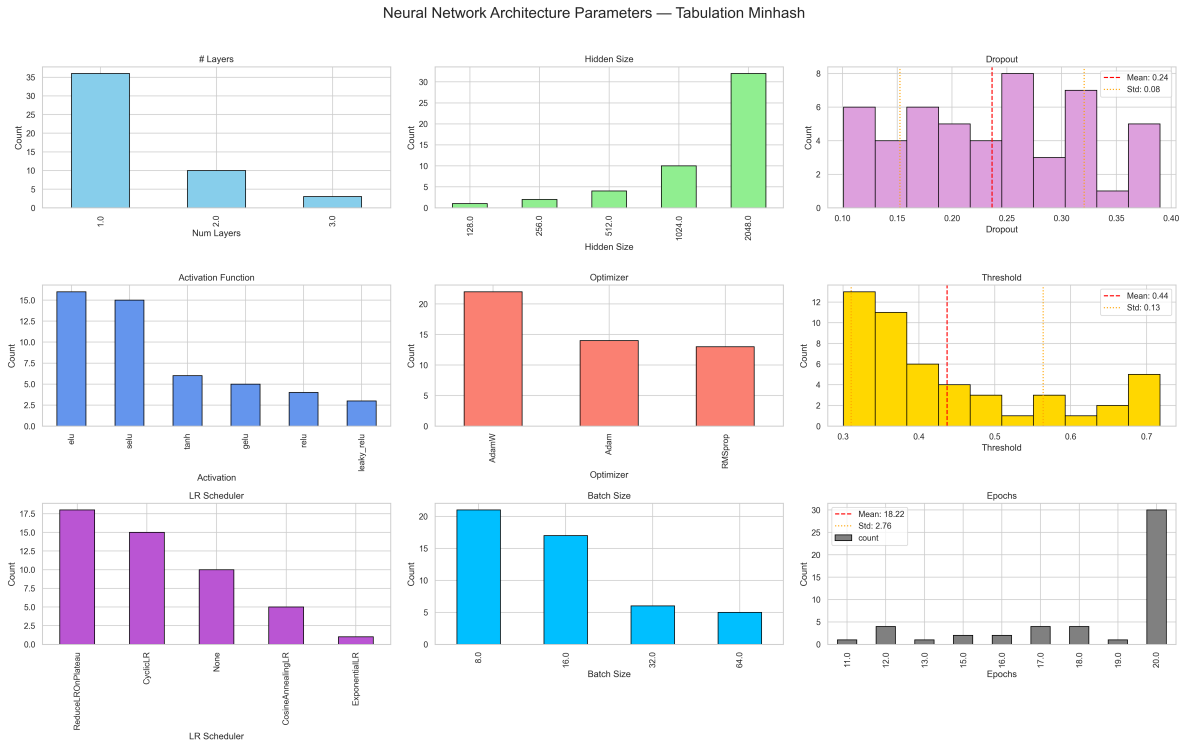


**Figure A.9:** Distribution of selected neural network architecture parameters during hyperparameter optimization for the Tabulation MinHash encoding.

## A.2 Two-Step Hash: Dataset Extension Attack Results

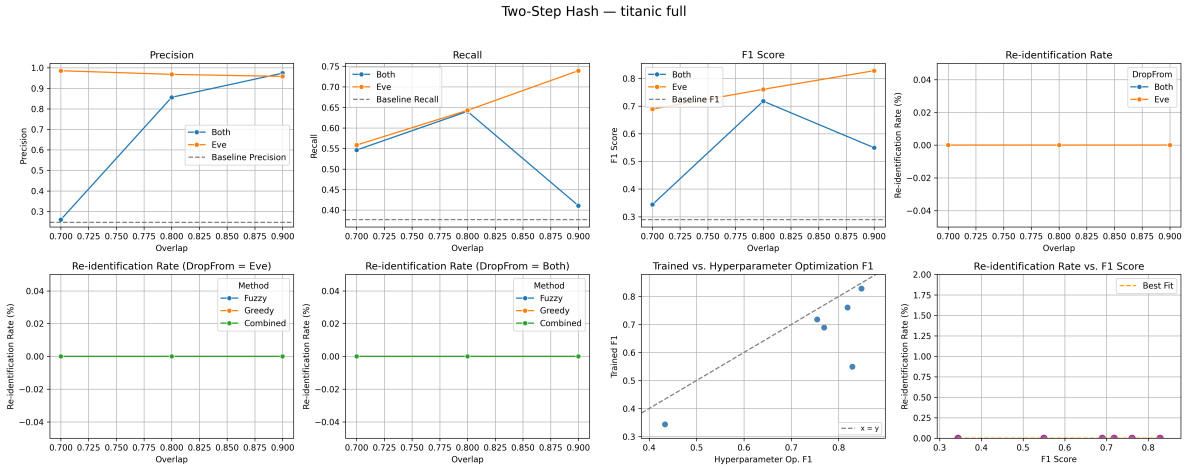


**Figure A.10:** Two-Step Hash results on the `titanic_full` dataset.

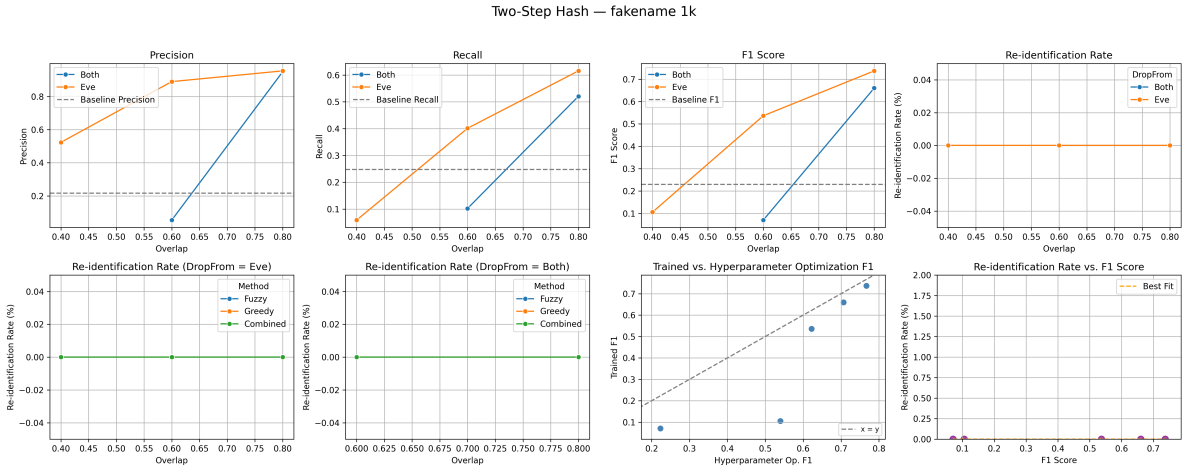


**Figure A.11:** Two-Step Hash results on the `fakename_1k` dataset.

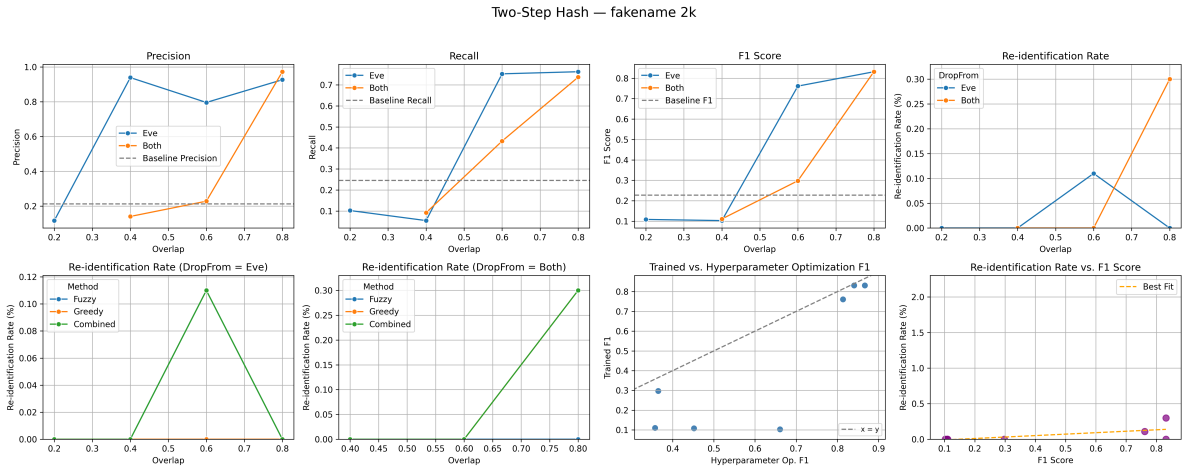


**Figure A.12:** Two-Step Hash results on the `fakename_2k` dataset.

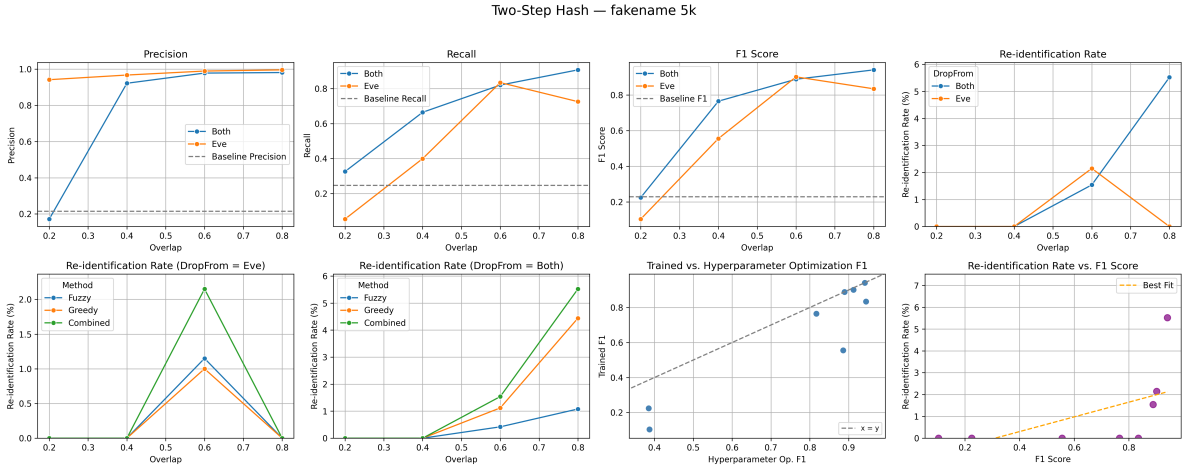


**Figure A.13:** Two-Step Hash results on the `fakename_5k` dataset.

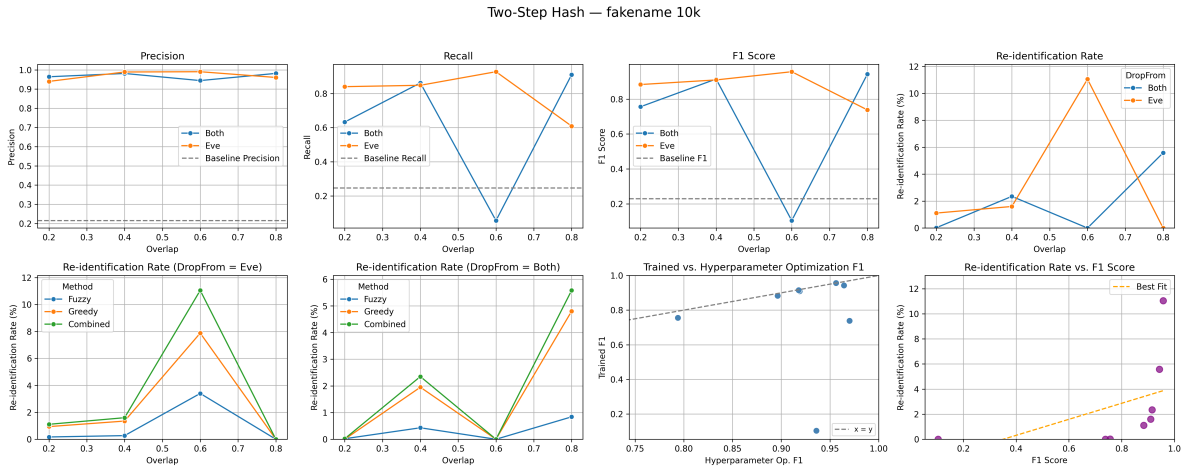


**Figure A.14:** Two-Step Hash results on the `fakename_10k` dataset.

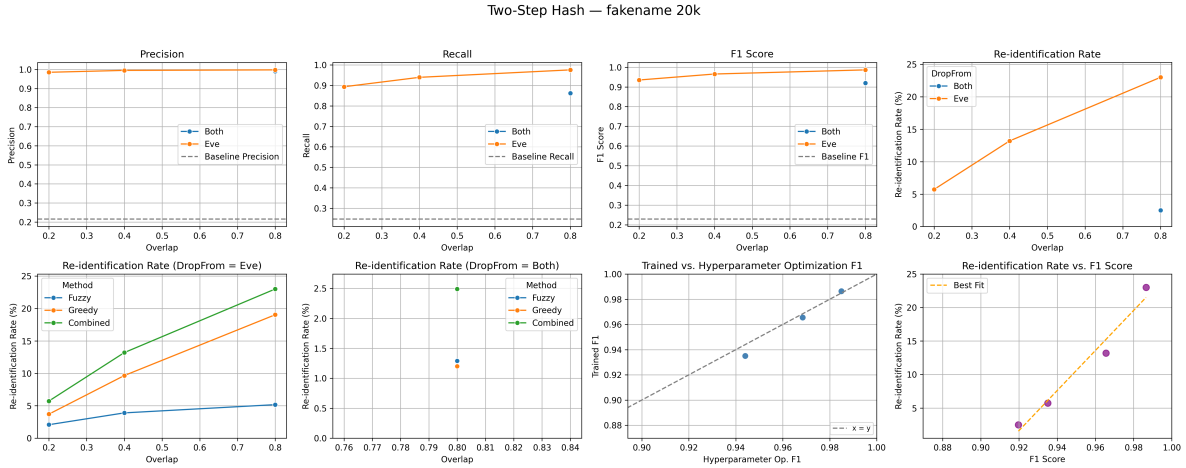


**Figure A.15:** Two-Step Hash results on the `fakename_20k` dataset.

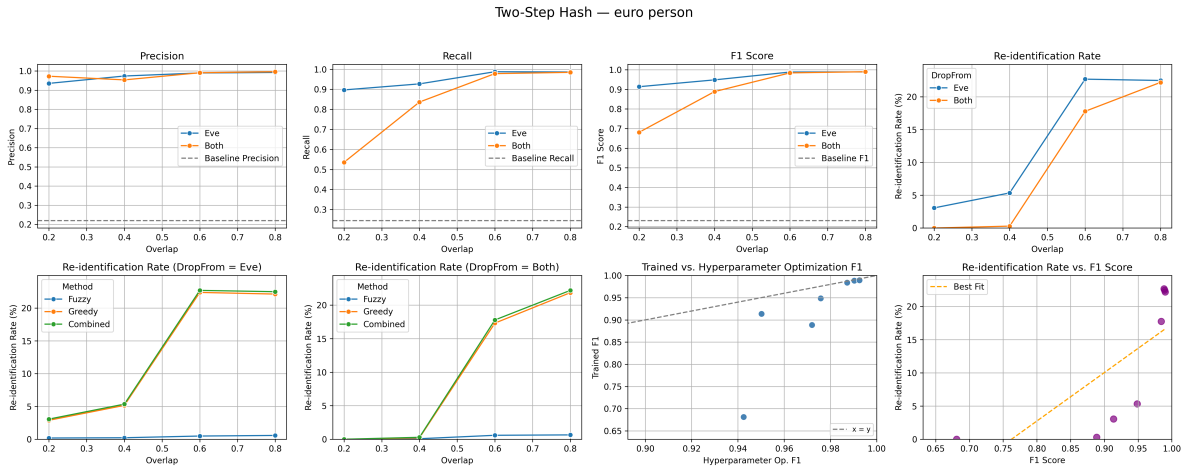


**Figure A.16:** Two-Step Hash results on the `euro_person` dataset.

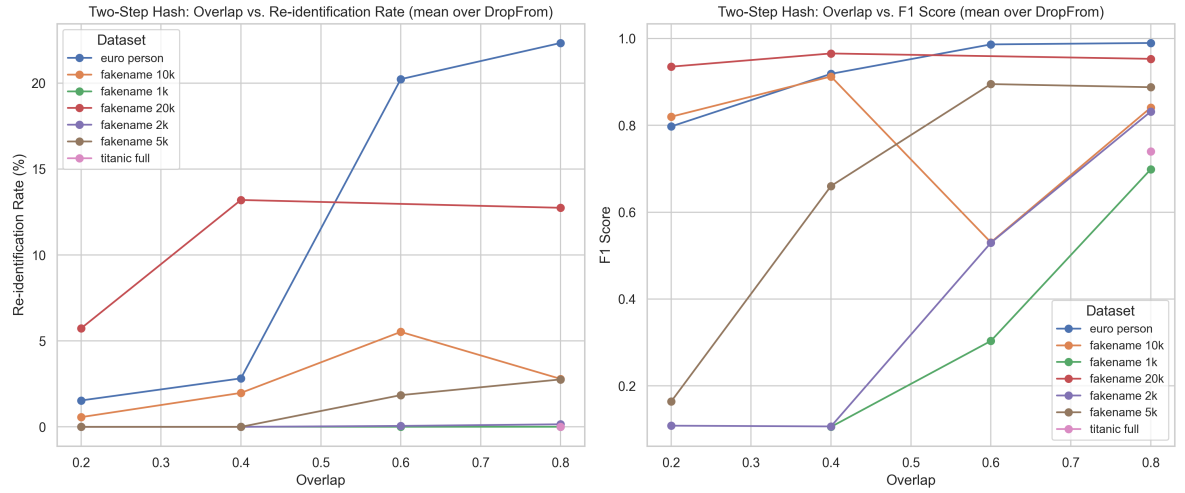


**Figure A.17:** Comparison of re-identification rates and F1 scores across all datasets with Two-Step Hash encoding as a function of overlap.

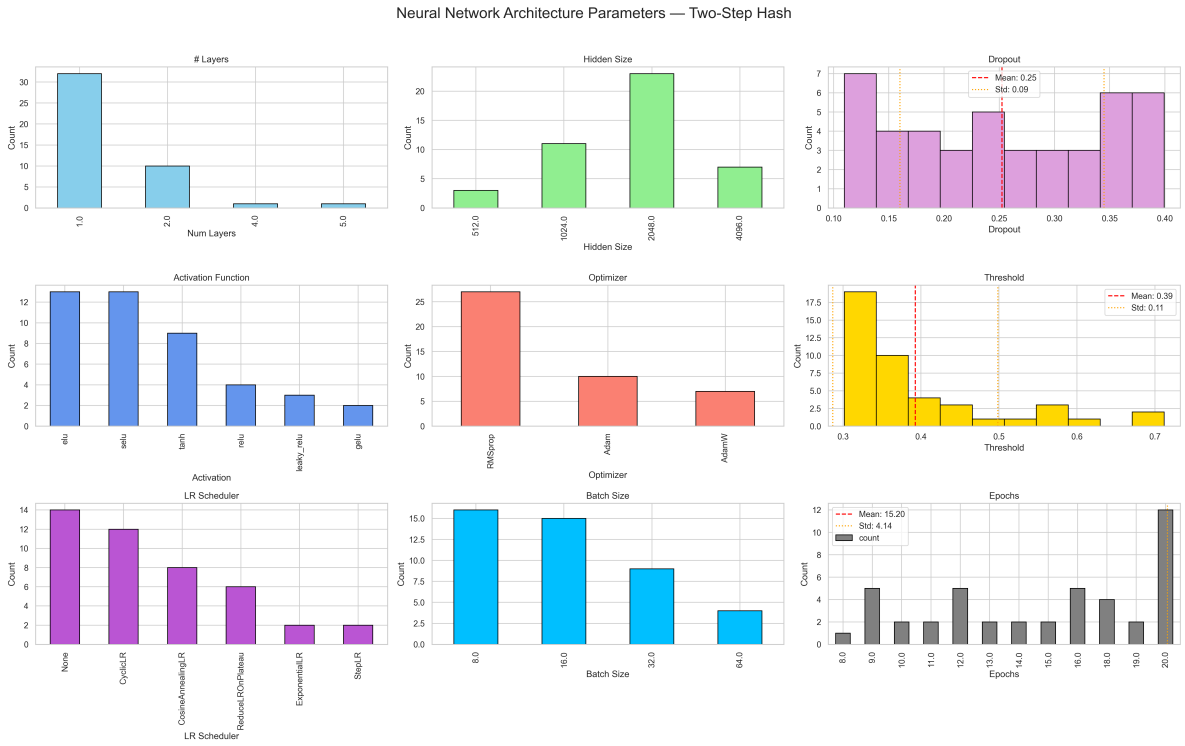


**Figure A.18:** Distribution of selected neural network architecture parameters during hyperparameter optimization for the Two-Step Hash encoding.

## A.3 Bloom Filter: Dataset Extension Attack Results

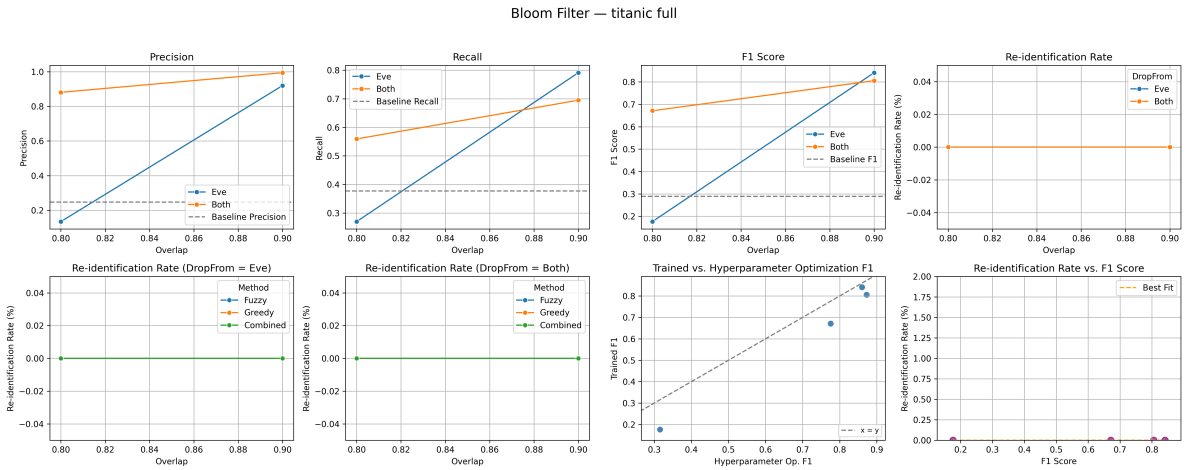


**Figure A.19:** Bloom Filter results on the `titanic_full` dataset.

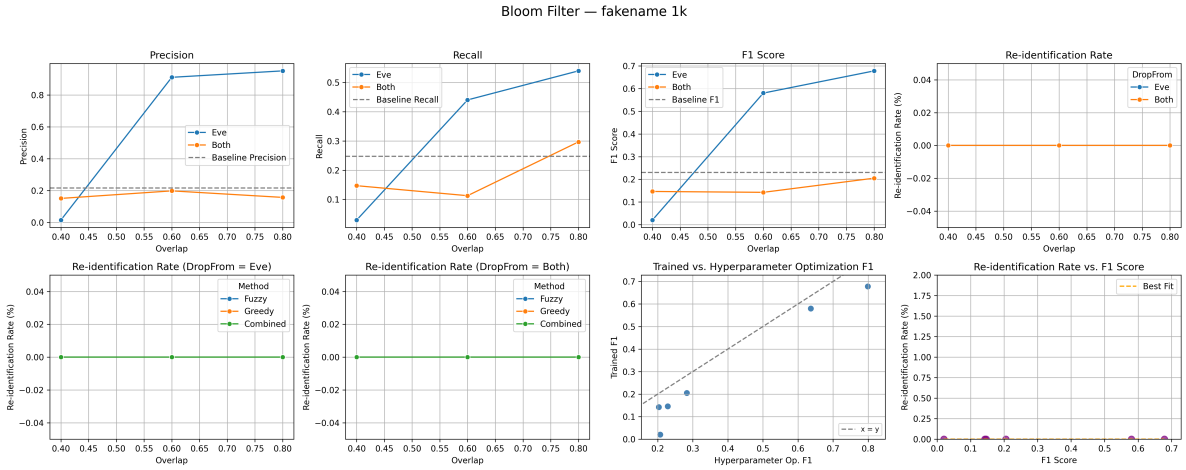


**Figure A.20:** Bloom Filter results on the `fakename_1k` dataset.

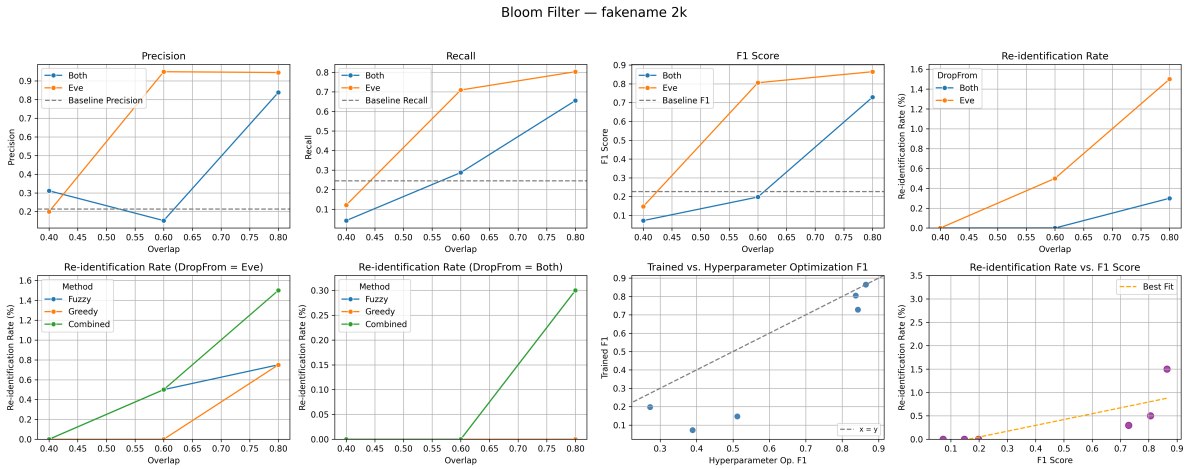


**Figure A.21:** Bloom Filter results on the `fakename_2k` dataset.

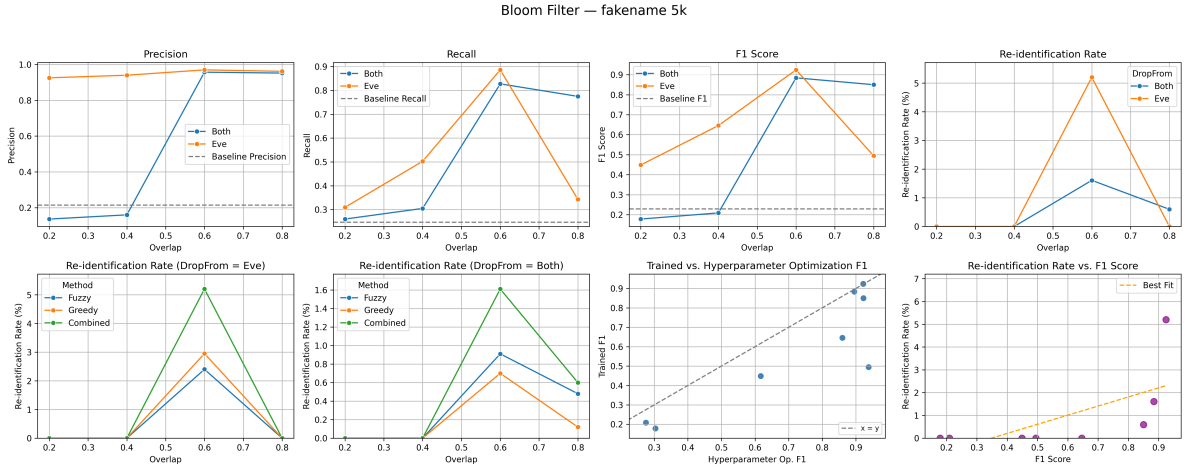


**Figure A.22:** Bloom Filter results on the `fakename_5k` dataset.

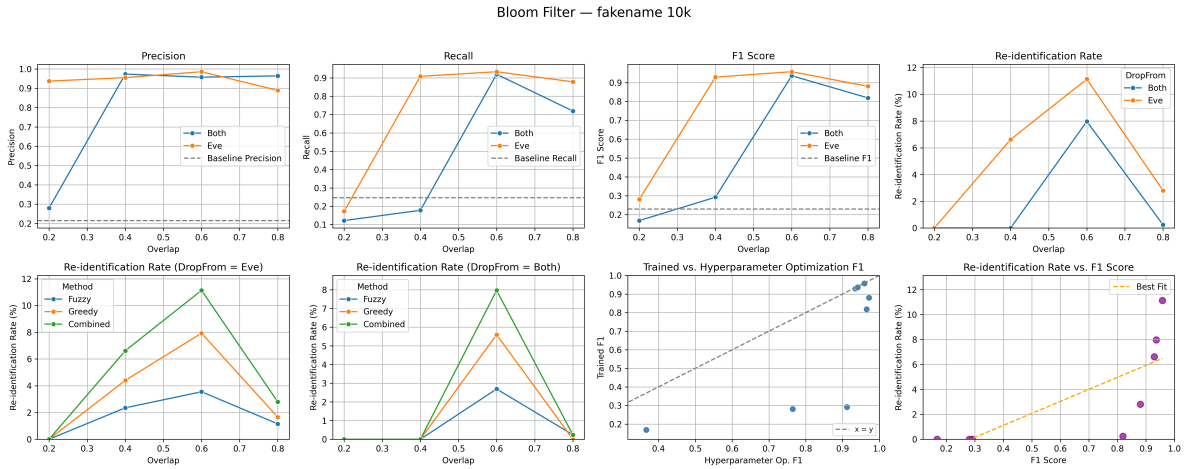


**Figure A.23:** Bloom Filter results on the `fakename_10k` dataset.

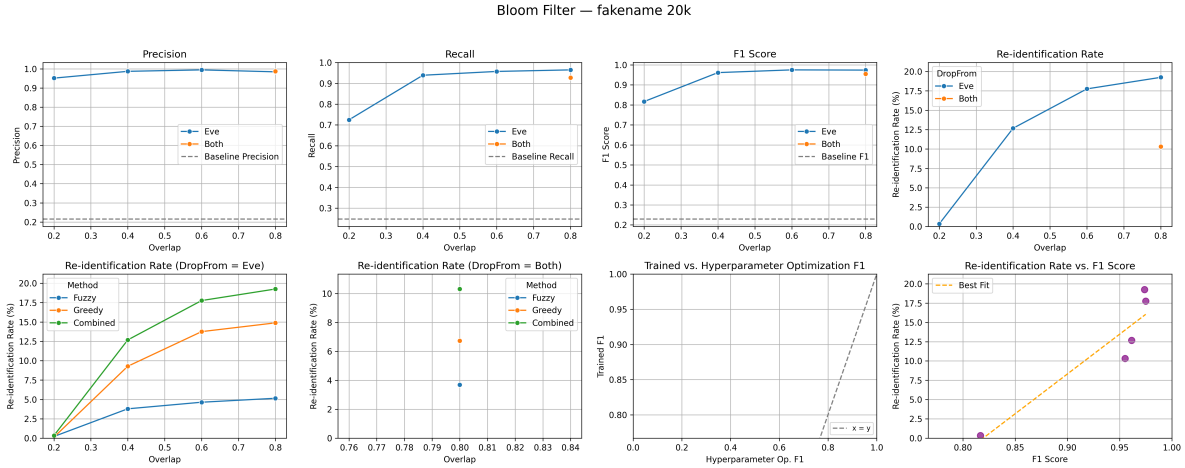


**Figure A.24:** Bloom Filter results on the `fakename_20k` dataset.

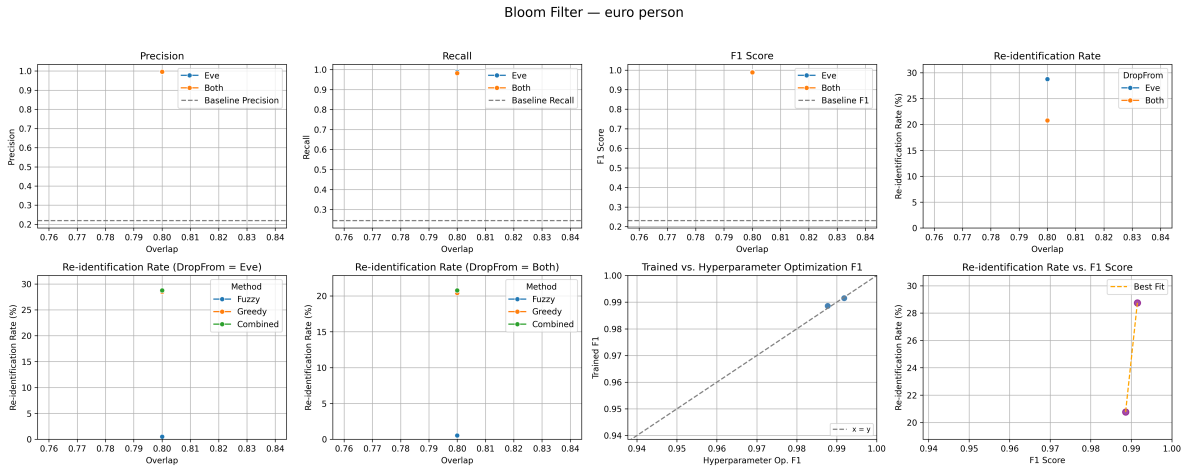


**Figure A.25:** Bloom Filter results on the `euro_person` dataset.

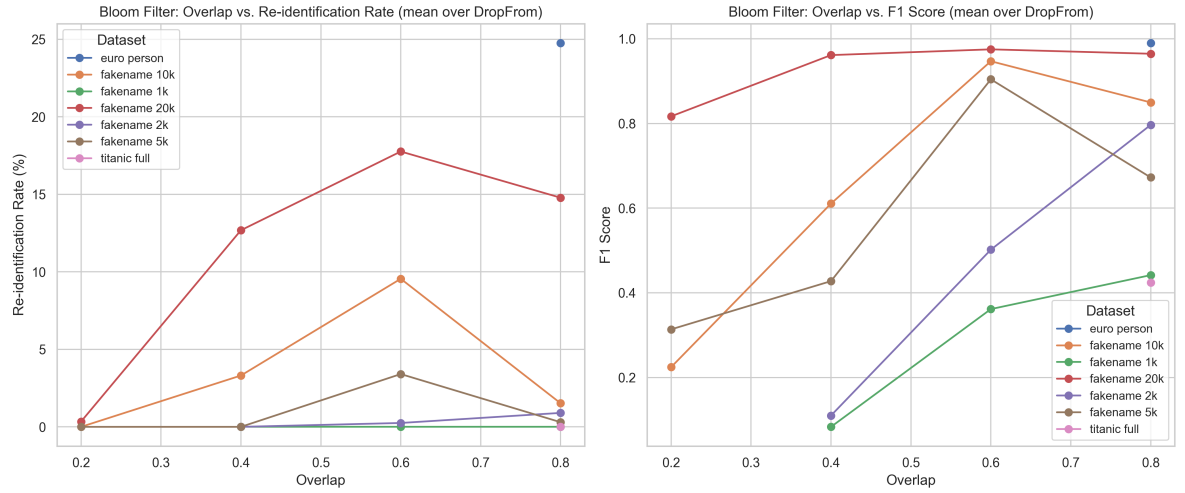


**Figure A.26:** Comparison of re-identification rates and F1 scores across all datasets with Bloom Filter encoding as a function of overlap.

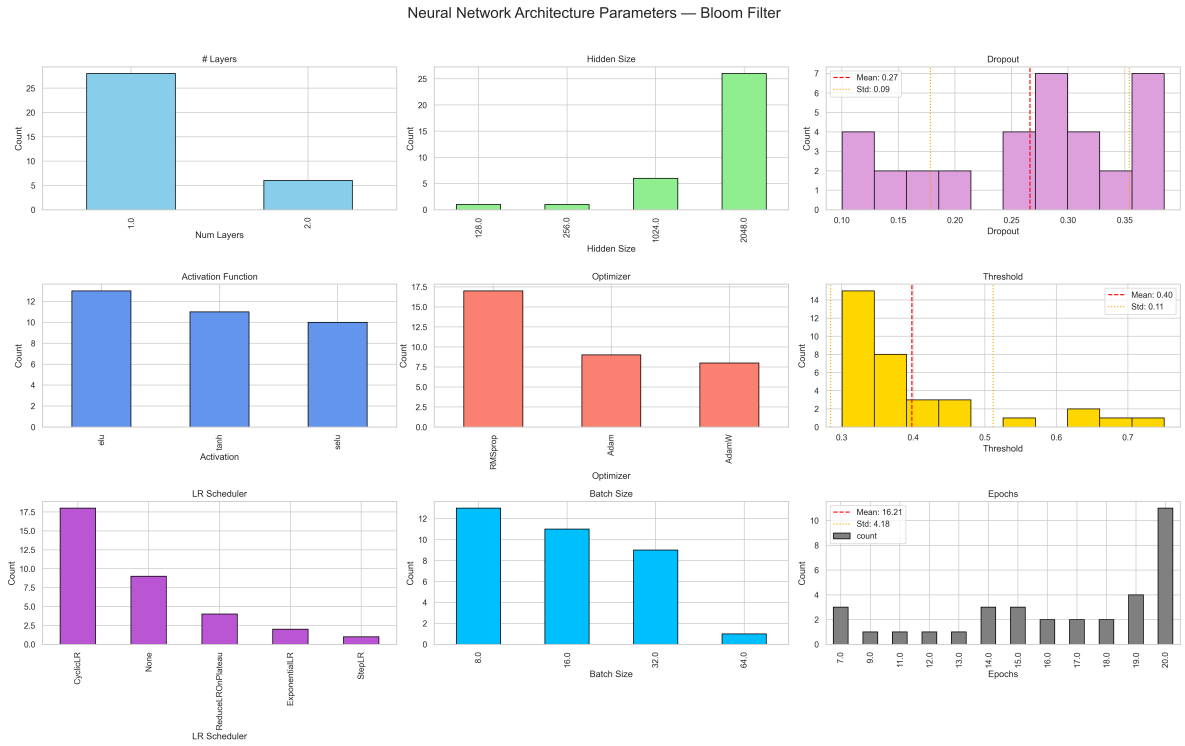


**Figure A.27:** Distribution of selected neural network architecture parameters during hyperparameter optimization for the Bloom Filter encoding.

## A.4 Encoding Scheme Comparison: Dataset Extension Attack Results

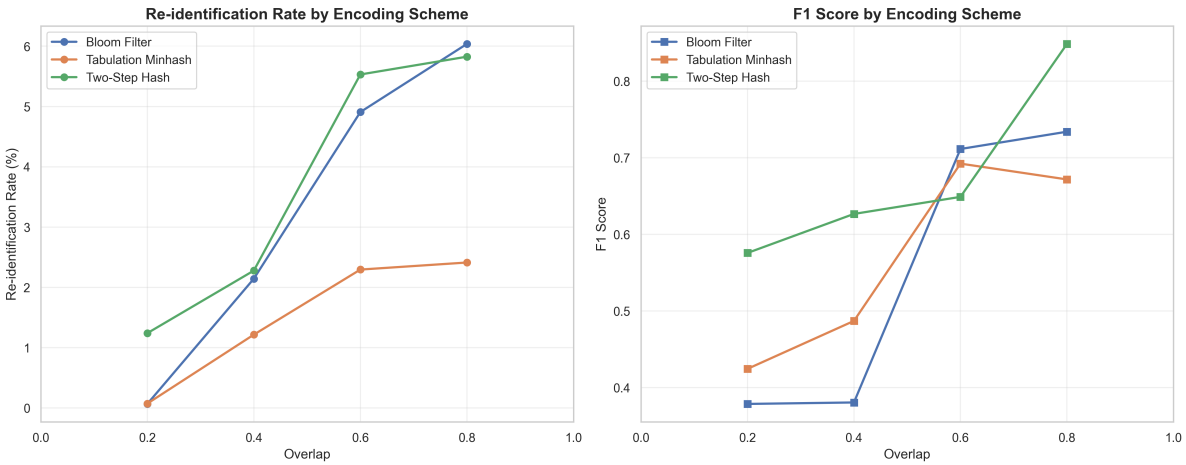


**Figure A.28:** Line plots of mean re-identification rate and F1 score across encoding schemes as a function of overlap.

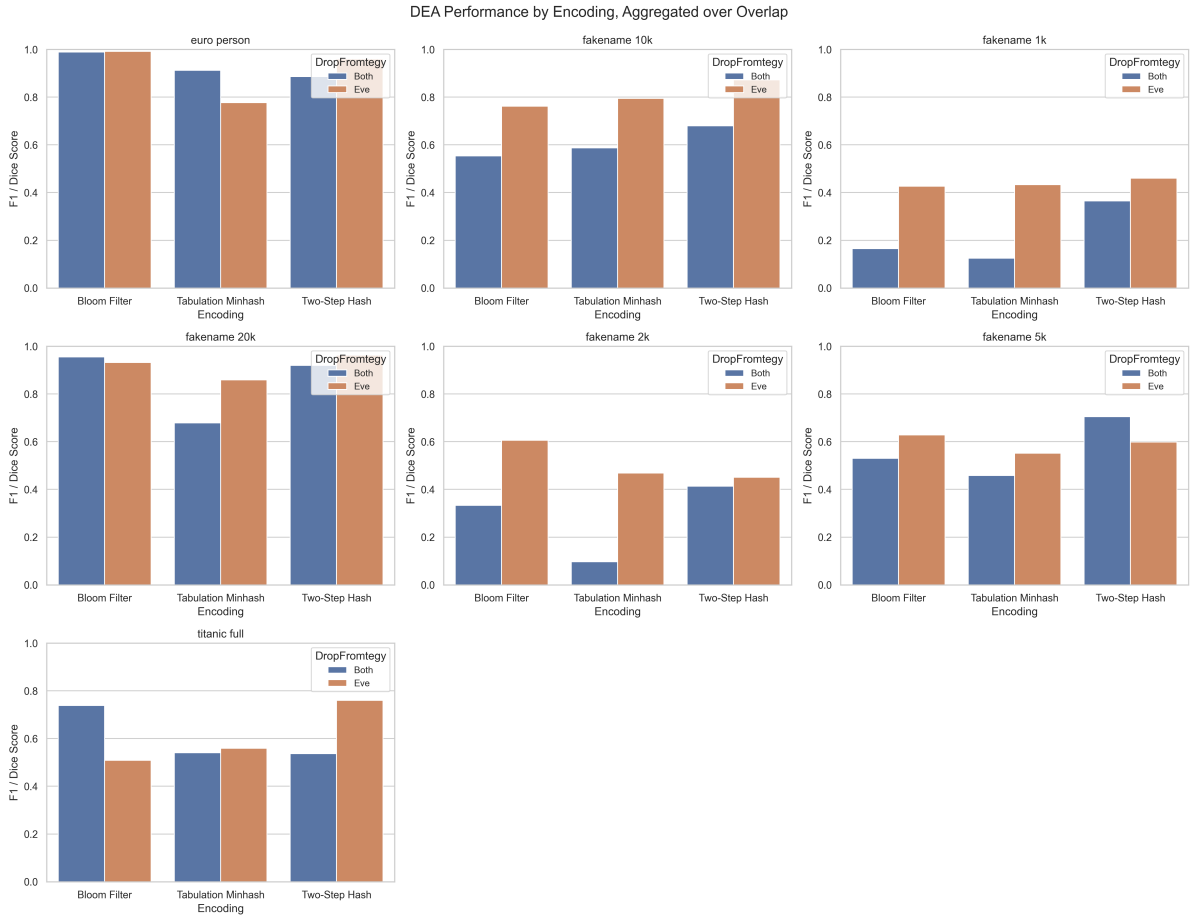


**Figure A.29:** Comparison of Dataset Extension Attack F1 scores for Bloom Filter, Tabulation MinHash, and Two-Step Hash across all datasets, averaged over overlap values and separated by DropFrom strategy (Eve vs. Both).

# Eidesstattliche Erklärung

Hiermit versichere ich, dass diese Arbeit von mir persönlich verfasst wurde und dass ich keinerlei fremde Hilfe in Anspruch genommen habe. Ebenso versichere ich, dass diese Arbeit oder Teile daraus weder von mir selbst noch von anderen als Leistungsnachweise andernorts eingereicht wurden. Wörtliche oder sinngemäße Übernahmen aus anderen Schriften und Veröffentlichungen in gedruckter oder elektronischer Form sind gekennzeichnet. Sämtliche Sekundärliteratur und sonstige Quellen sind nachgewiesen und in der Bibliographie aufgeführt. Das Gleiche gilt für graphische Darstellungen und Bilder sowie für alle Internet-Quellen. Ich bin ferner damit einverstanden, dass meine Arbeit zum Zwecke eines Plagiatsabgleichs in elektronischer Form anonymisiert versendet und gespeichert werden kann. Mir ist bekannt, dass von der Korrektur der Arbeit abgesehen werden kann, wenn diese Erklärung nicht erteilt wird.

| | |
|---|---|
| Datum | Marcel Mildenberger |